Attorney Docket No. YOR920000581US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s):	J.L. Hellerstein et al.
Docket No.:	YOR920000581US1
Serial No.:	09/731,937
Filing Date:	December 7, 2000
Group:	2144
Examiner:	Thanh T. Nguyen
Title:	Method and System for Machine-Aided Rule Construction for Event Management

---

APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Applicants (hereinafter "Appellants") hereby appeal the final rejection dated June 15, 2007 of claims 1-20 of the above-identified application.

REAL PARTY IN INTEREST

The present application is assigned of record to International Business Machines Corporation, which is the real party in interest.

RELATED APPEALS AND INTERFERENCES

There are no known related appeals or interferences.

STATUS OF CLAIMS

The present application was filed on December 7, 2000 with claims 1-20, all of which remain pending. Claims 1, 8 and 14 are the independent claims.

Claims 1-20 stand finally rejected under 35 U.S.C. §103(a). Claims 1-20 are appealed.

STATUS OF AMENDMENTS

There have been no amendments filed subsequent to the final rejection.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a computer-based method of constructing one or more correlation rules for use by an event management system for managing a network with one or more computing devices. The method includes the steps of selecting one or more event patterns representing event data associated with the network of computing devices being managed by the event management system; automatically learning predicates of the one or more correlation rules from the one or more selected event patterns; and adding one or more corresponding actions to the one or more automatically learned predicates to form the one or more correlation rules.

As discussed with reference to FIG. 1 in the specification at page 9, line 15, to page 10, line 19, an illustrative embodiment of the method recited in independent claim 1 is directed to a computer-based method of constructing one or more correlation rules (e.g., stored in Rule DB 185) for use by an event management system (e.g, 110) for managing a network (e.g., 115) with one or more computing devices (e.g., file servers 132, name servers 134, mail servers 136, etc.). As discussed with reference to FIG. 3 in the specification at page 11, line 22, to page 12, line 16, the method includes the steps of selecting one or more event patterns representing event data associated with the network of computing devices being managed by the event management system (e.g., step 405); automatically learning predicates of the one or more correlation rules from the one or more selected event patterns (e.g., step 410); and adding one or more corresponding actions to the one or more automatically learned predicates to form the one or more correlation rules (e.g., step 420).

Independent claim 8 is directed to an apparatus for constructing one or more correlation rules for use by an event management system for managing a network with one or more computing devices. The apparatus comprises at least one processor and a memory, coupled to the at least one processor, which stores the one or more correlation rules for access by the event management system. The at least one processor is operative to (i) permit selection of one or more event patterns representing event data associated with the network of computing devices being managed by the event management system; (ii) automatically learn predicates of the one or more correlation rules from the one or more selected event patterns; and (iii) add one or more corresponding actions to the one or more automatically learned predicates to form the one or more correlation rules.

As discussed with reference to FIG. 1 in the specification at page 9, line 15, to page 10, line 19, an illustrative embodiment of the apparatus recited in independent claim 8 is directed to an apparatus for constructing one or more correlation rules (e.g., stored in Rule DB 185) for use by an event management system (e.g., 110) for managing a network (e.g., 115) with one or more computing devices (e.g., file servers 132, name servers 134, mail servers 136, etc.). As discussed with reference to FIG. 11 in the specification at page 18, line 3, to page 19, line 3, the apparatus comprises at least one processor (e.g., 1100) and a memory (e.g., 1110), coupled to the at least one processor, which stores the one or more correlation rules for access by the event management system. As discussed with reference to FIG. 3 in the specification at page 11, line 22, to page 12, line 16, the at least one processor is operative to (i) permit selection of one or more event patterns representing event data associated with the network of computing devices being managed by the event management system (e.g., step 405); (ii) automatically learn predicates of the one or more correlation rules from the one or more selected event patterns (e.g., step 410); and (iii) add one or more corresponding actions to the one or more automatically learned predicates to form the one or more correlation rules (e.g., step 420).

Independent claim 14 is directed to an article of manufacture for constructing one or more correlation rules for use by an event management system for managing a network with one or more computing devices. The article comprises a machine readable medium containing one or more programs which when executed implement at least one of the steps of: selecting one or

more event patterns representing event data associated with the network of computing devices being managed by the event management system; automatically learning predicates of the one or more correlation rules from the one or more selected event patterns; and adding one or more corresponding actions to the one or more automatically learned predicates to form the one or more correlation rules.

As discussed with reference to FIG. 1 in the specification at page 9, line 15, to page 10, line 19, and with reference to FIG. 11 in the specification at page 18, line 3, to page 19, line 3, an illustrative embodiment of the article of manufacture recited in independent claim 1 is directed to constructing one or more correlation rules (e.g., stored in Rule DB 185) for use by an event management system (e.g, 110) for managing a network (e.g., 115) with one or more computing devices (e.g., file servers 132, name servers 134, mail servers 136, etc.). As discussed with reference to FIG. 3 in the specification at page 11, line 22, to page 12, line 16, the article comprises a machine readable medium containing one or more programs which when executed implement at least one of the steps of: selecting one or more event patterns representing event data associated with the network of computing devices being managed by the event management system (e.g., step 405); automatically learning predicates of the one or more correlation rules from the one or more selected event patterns (e.g., step 410); and adding one or more corresponding actions to the one or more automatically learned predicates to form the one or more correlation rules (e.g., step 420).

As discussed in the present specification at, for example, page 7, line 19, to page 8, line 4, illustrative embodiments of the present invention advantageously permit easier construction of correlation rules and superior assessment of correlation rules once they are constructed.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-20 stand rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,446,136 to Pohlmann et al. (hereinafter "Pohlmann") in view of U.S. Patent No. 6,584,186 to Aravamudan et al. (hereinafter "Aravamudan").

ARGUMENT

Claims 1, 2, 4, 8, 10, 14, 15 and 17

Appellants respectfully submit that the Examiner has failed to establish a proper *prima facie* case of obviousness in the §103(a) rejection of independent claim 1, for at least the reasons that the combined teachings of Pohlmann and Aravamudan fail to teach or suggest all limitations of claim 1 and that no cogent motivation has been identified for combining or modifying the reference teachings to reach the claimed invention.

The combination of Pohlmann and Aravamudan fails to disclose the element of automatically learning predicates of the correlation rules from selected event patterns. In formulating the rejection of independent claims 1, 8 and 14 on pages 3, 5 and 7 of the present Office Action, respectively, the Examiner contends that this limitation is disclosed by Pohlmann at column 5, lines 33-35; column 5, lines 45-51; and column 6, lines 14-19. Appellants respectfully submit that the portions of Pohlmann cited by the Examiner disclose the querying of events and the forwarding or publishing of events matching a subscription request to the requestor. Pohlmann fails to disclose anything regarding predicates of correlation rules, or the automatic learning of such predicates from selected event patterns.

On paragraph 27 found on page 9 of the present Office Action, the Examiner responds to Appellants' previous arguments by alternatively contending that this limitation is "clearly shown" in Aravamudan at column 14, lines 15-35. However, Appellants respectfully submit that Aravamudan fails to remedy the deficiencies described above with regard to Pohlmann. More specifically, the portion of Aravamudan referred to by the Examiner discloses the correlation of two events in order to issue new policies to reprogram network elements. Aravamudan fails to specifically disclose that predicates of correlation rules are automatically learned from selected event patterns. Aravamudan fails to "clearly show predicates of the correlation rules for selected event patterns," as the Examiner contends.

Moreover, the Examiner contends that "since Aravamudan does not mention or indicate about manually learning predicate, therefore, it is automatically learning," without providing evidence that supports such a conclusion. The Examiner appears to be suggesting that Aravamudan inherently discloses automatic learning. Appellants respectfully note that the

Federal Circuit has held that inherency "may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.'" *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999) (citations omitted). The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. *In re Rijckaert*, 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993) (reversed rejection because inherency was based on what would result due to optimization of conditions, not what was necessarily present in the prior art); *In re Oelrich*, 666 F.2d 578, 581-82, 212 USPQ 323, 326 (CCPA 1981).

Rather, "the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art." *Ex parte Levy,* 17 USPQ2d 1461, 1464 (Bd. Pat. App. & Inter. 1990) (emphasis in original). Appellants respectfully submit that the Examiner has failed to provide any evidence to support the conclusion that "since Aravamudan does not mention or indicate about manually learning predicate," it necessarily flows that Aravamudan discloses automatic learning as recited in claim 1.

The combination of Pohlmann and Aravamudan also fails to disclose the element of adding corresponding actions to the automatically learned predicates to form correlation rules. The Examiner concedes that this limitation is neither taught nor suggested by Pohlmann. Instead, the Examiner contends that this element is provided in column 14, lines 15-35 of Aravamudan. As discussed above, however, this portion of Aravamudan cited by the Examiner discloses the correlation of two events in order to issue new policies to reprogram network elements. Aravamudan fails to disclose anything regarding the addition of corresponding actions to automatically learned predicates to form correlation rules. Pohlmann fails to remedy the deficiencies described above with regard to Aravamudan. Therefore, the combination of Pohlmann and Aravamudan fails to disclose this element of the independent claims.

Moreover, Appellants respectfully submit that Aravamudan is not analogous prior art and therefore cannot form the basis for a rejection under 35 U.S.C. §103. See, e.g., MPEP §2141.01(a); *In re Oetiker*, 977 F.2d 1443, 1446, 24 USPQ2d 1443, 1445 (Fed. Cir. 1992) ("In order to rely on a reference as a basis for rejection of an applicant's invention, the reference must

either be in the field of applicant's endeavor or, if not, then be reasonably pertinent to the particular problem with which the inventor was concerned."); *In re Clay*, 966 F.2d 656, 659, 23 USPQ2d 1058, 1060-61 (Fed. Cir. 1992) ("A reference is reasonably pertinent if, even though it may be in a different field from that of the inventor's endeavor, it is one which, because of the matter with which it deals, logically would have commended itself to an inventor's attention in considering his problem.").

Whereas Pohlmann is directed to an event management system in which an event manager provides and receives events, an event correlator correlates at least one of the events based on alarm rules, and a response engine executes a response policy based on the correlation of events by the event correlator, Aravamudan is directed to methods and apparatus for protecting against network damage in next generation communication networks. As such, Appellants respectfully submit that Aravamudan is thus neither in the field of Appellants' endeavor nor logically would have commended itself to an inventor's attention in considering his problem, much less have been an obvious candidate for combination with Pohlmann.

In the final Office Action, on pages 3-4, the Examiner provides the following statement to prove motivation to combine Pohlmann and Aravamudan, with emphasis supplied:

> It would have been obvious . . . to combine the teachings of <u>Pohlmann and Aravamudan</u> to have one or more corresponding actions to the one or more automatically learn predicates to form the one or more correlation rules <u>because it would have allowed providers to perform automatic dynamic market testing and automatically adjusted served content based on responses from users</u>.

Appellants respectfully submit that the above-quoted explanation is a conclusory statement of the sort rejected by both the Federal Circuit and the U.S. Supreme Court. See *KSR v. Teleflex*, 127 S.Ct. 1727, 1741, 82 USPQ2d 1385, 1396 (U.S., Apr. 30, 2007), quoting *In re Kahn*, 441 F. 3d 977, 988 (Fed. Cir. 2006) ("[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness."). There has been no showing in the present §103(a) rejection of claim 1 of objective evidence of record that would motivate one skilled in the art to combine Pohlmann with Aravamudan to produce the particular

limitations in question. Rather, the above-quoted statements of motivation provided by the Examiner are conclusory statements of the type ruled insufficient in the *KSR* case.

More specifically, the statement above is using the benefit allegedly obtained from a combination as a motivation for that combination; this is impermissible hindsight. "It is improper, in determining whether a person of ordinary skill would have been led to this combination of references, simply to '[use] that which the inventor taught against its teacher.'" *In re Sang-Su Lee*, 277 F.3d 1338, 1344 (Fed. Cir. 2002) (quoting *W.L. Gore v. Garlock, Inc.*, 721 F.2d 1540, 1553, 220 USPQ 303, 312-13 (Fed. Cir. 1983)).

Instead, in order to avoid the improper use of a hindsight-based obviousness analysis, particular findings must be made as to why one skilled in the relevant art, having no knowledge of the claimed invention, would have combined the teachings of Pohlmann with those of Aravamudan in the claimed manner. See, e.g., *In re Kotzab*, 217 F.3d 1365, 1371, 55 USPQ2d 1313, 1317 (Fed. Cir. 2000).

Appellants also note that the aforementioned statement of motivation is identical to that provided by the Examiner with regard to the motivation to combine Pohlmann with U.S. Patent No. 6,477,575 (hereinafter "Koeppel"). See, for example, the previous Office Actions dated April 23, 2004; February 17, 2005; October 20, 2005; April 13, 2006, each of which state, with emphasis supplied:

> It would have been obvious to one of ordinary skill in the art at the time of the invention was made to combine the teachings of <u>Pohlmann and Koeppel</u> to have one or more corresponding actions to the one or more automatically learn predicates to form the one or more correlation rules <u>because it would have allowed providers to perform automatic dynamic market testing and automatically adjusted served content based on responses from users (see Koeppel's col. 14, lines 38-62)</u>.

See also the present Office Action at page 9, first paragraph, and the present Advisory Action at page 2, second paragraph, each of which state, with emphasis supplied:

> In response to applicant's argument that there is no suggestion to combine <u>Pohlmann and Koeppel</u> the examiner recognizes that obviousness can only be

> established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion , or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. . . . In this case, <u>it would have allowed providers to perform automatic dynamic market testing and automatically adjusted served content based on responses from users.</u>

The Koeppel reference was removed from consideration pursuant to 37 CFR 1.131 based on the Declarations and Affidavit included in the present Evidence Appendix, which established that the inventors conceived an invention within the scope of the claims prior to the effective date of the Koeppel reference. Accordingly, the Examiner's continued reliance on the alleged teachings of the Koeppel reference as the reason why one would have been motivated to combine Pohlmann with Aravamudan at the time the invention was made constitutes impermissible hindsight-based reasoning.

It is therefore believed that independent claim 1 is not obvious in view of the proposed combination of Pohlmann and Aravamudan.

Independent claims 8 and 14 contain limitations similar to those of independent claim 1 and are thus allowable for reasons similar to those identified above with regard to claim 1.

Dependent claims 2, 4, 10, 15 and 17 are believed allowable for at least the reasons identified above with regard to their respective independent claims.

<u>Claims 3, 9 and 16</u>

Dependent claims 3, 9 and 16 are believed allowable for at least the reasons identified above with regard to their respective independent claims. Moreover, these claims contain independently patentable subject matter. Specifically, each of these claims contains similar limitations wherein event pattern selection comprises a user marking the one or more event patterns in accordance with a data visualization of at least a portion of the event data.

The Examiner contends that this limitation is taught by Pohlmann at column 4, lines 27-35. Appellants respectfully submit that the relied-upon portion of Pohlmann, which is directed toward self-contained discrete events, fails to teach or suggest the aforementioned limitations

directed toward a user marking the one or more event patterns in accordance with a data visualization of at least a portion of the event data.

Aravamudan fails to remedy the deficiencies described above with regard to Pohlmann. Therefore, the combination of Pohlmann and Aravamudan fails to disclose the additional limitations of dependent claims 3, 9 and 16.

Claims 5, 11 and 18

Dependent claims 5, 8 and 18 are believed allowable for at least the reasons identified above with regard to their respective independent claims. Moreover, these claims contain independently patentable subject matter. For example, each of these claims contains similar limitations wherein automatic predicate learning comprises learning an initial concept.

In formulating the rejections of these dependent claims in the final Office Action, the Examiner fails to indicate the portion of Pohlmann which are believed to teach or suggest the aforementioned limitation wherein automatic predicate learning comprises learning an initial concept. Appellants respectfully submit that the Examiner has failed to comply with 37 CFR 1.104(c)(2) ("In rejecting claims for want of novelty or for obviousness, the examiner must cite the best references at his or her command. When a reference is complex or shows or describes inventions other than that claimed by the applicant, the particular part relied on must be designated as nearly as practicable. The pertinence of each reference, if not apparent, must be clearly explained and each rejected claim specified.")

Moreover, these rejections are not only procedurally deficient, but they are also substantively incorrect. Pohlmann fails to teach or suggest automatic predicate learning, much less learning an initial concept. Rather, Pohlmann teaches a technique involving the querying of events and the forwarding or publishing of events matching a subscription request to the requestor.

Aravamudan fails to remedy the deficiencies described above with regard to Pohlmann. Therefore, the combination of Pohlmann and Aravamudan fails to disclose the additional limitations of dependent claims 5, 8 and 18.

Claims 6, 12 and 19

Dependent claims 6, 12 and 19 are believed allowable for at least the reasons identified above with regard to their respective independent claims. Moreover, these claims contain independently patentable subject matter. Specifically, each of these claims contains similar limitations wherein automatic predicate learning utilizes one or more abstraction hierarchies.

The Examiner contends that this limitation is taught by Pohlmann at column 12, lines 46-50, and column 13, lines 14-20. Even assuming that the relied-upon portions of Pohlmann discloses the claimed abstraction hierarchies, the relied-upon portions of Pohlmann nonetheless fail to teach or suggest the aforementioned limitation wherein one or more abstraction hierarchies are utilized in automatic predicate learning. Rather, the relied-upon portions of Pohlmann disclose hierarchical consolidation on a subscription basis. See, e.g., Pohlmann at column 12, line 46.

Aravamudan fails to remedy the deficiencies described above with regard to Pohlmann. Therefore, the combination of Pohlmann and Aravamudan fails to disclose the additional limitations of dependent claims 6, 12 and 19.

Claims 7, 13 and 20

Dependent claims 7, 13 and 20 are believed allowable for at least the reasons identified above with regard to their respective independent claims, as well as their immediate base claims 6, 12 and 19. Moreover, these claims contain independently patentable subject matter. Specifically, each of these claims contains similar limitations wherein the one or more abstraction hierarchies comprise a hierarchy for at least one of a host and an event type.

The Examiner contends that this limitation is taught by Pohlmann at column 4, lines 27-45. Appellants respectfully submit that the relied-upon portions of Pohlmann fail to even mention any hierarchies, much less the limitations of claims 7, 13 and 20 directed toward abstraction hierarchies comprise a hierarchy for at least one of a host and an event type.

Aravamudan fails to remedy the deficiencies described above with regard to Pohlmann. Therefore, the combination of Pohlmann and Aravamudan fails to disclose the additional limitations of dependent claims 7, 13 and 20.

In view of the above, Appellants believe that claims 1-20 are in condition for allowance, and respectfully request withdrawal of the §103(a) rejection.

Respectfully submitted,

Date: December 17, 2007

William E. Lewis
Attorney for Appellant(s)
Reg. No. 39,274
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560
(516) 759-2946

CLAIMS APPENDIX

1. A computer-based method of constructing one or more correlation rules for use by an event management system for managing a network with one or more computing devices, the method comprising the steps of:

selecting one or more event patterns representing event data associated with the network of computing devices being managed by the event management system;

automatically learning predicates of the one or more correlation rules from the one or more selected event patterns; and

adding one or more corresponding actions to the one or more automatically learned predicates to form the one or more correlation rules.

2. The method of claim 1, further comprising the step of storing the one or more correlation rules in a rule database for access by the event management system.

3. The method of claim 1, wherein the event pattern selection step further comprises the step of a user marking the one or more event patterns in accordance with a data visualization of at least a portion of the event data.

4. The method of claim 1, wherein the event pattern selection step employs a data mining algorithm.

5. The method of claim 1, wherein the automatic predicate learning step comprises the steps of:

learning an initial concept;

determining if acceptance criteria are met given the event data;

querying historical event data for similar event patterns; and

allowing the user to edit the initial concept based on the historical event data query.

6. The method of claim 1, wherein the automatic predicate learning step utilizes one or more abstraction hierarchies.

7. The method of claim 6, wherein the one or more abstraction hierarchies comprise a hierarchy for at least one of a host and an event type.

8. Apparatus for constructing one or more correlation rules for use by an event management system for managing a network with one or more computing devices, the apparatus comprising:

at least one processor operative to:

(i) permit selection of one or more event patterns representing event data associated with the network of computing devices being managed by the event management system;

(ii) automatically learn predicates of the one or more correlation rules from the one or more selected event patterns; and

(iii) add one or more corresponding actions to the one or more automatically learned predicates to form the one or more correlation rules; and

a memory, coupled to the at least one processor, which stores the one or more correlation rules for access by the event management system.

9. The apparatus of claim 8, wherein the event pattern selection operation further comprises a user marking the one or more event patterns in accordance with a data visualization of at least a portion of the event data.

10. The apparatus of claim 8, wherein the event pattern selection operation employs a data mining algorithm.

11. The apparatus of claim 8, wherein the automatic predicate learning operation further comprises:

(i) learning an initial concept;

(ii) determining if acceptance criteria are met given the event data;

(iii) querying historical event data for similar event patterns; and

(iv) allowing the user to edit the initial concept based on the historical event data query.

12. The apparatus of claim 8, wherein the automatic predicate learning operation utilizes one or more abstraction hierarchies.

13. The apparatus of claim 12, wherein the one or more abstraction hierarchies comprise a hierarchy for at least one of a host and an event type.

14. An article of manufacture for constructing one or more correlation rules for use by an event management system for managing a network with one or more computing devices, the article comprising a machine readable medium containing one or more programs which when executed implement at least one of the steps of:

selecting one or more event patterns representing event data associated with the network of computing devices being managed by the event management system;

automatically learning predicates of the one or more correlation rules from the one or more selected event patterns; and

adding one or more corresponding actions to the one or more automatically learned predicates to form the one or more correlation rules.

15. The article of claim 14, further comprising the step of storing the one or more correlation rules in a rule database for access by the event management system.

16. The article of claim 14, wherein the event pattern selection step further comprises the

step of a user marking the one or more event patterns in accordance with a data visualization of at least a portion of the event data.

17. The article of claim 14, wherein the event pattern selection step employs a data mining algorithm.

18. The article of claim 14, wherein the automatic predicate learning step comprises the steps of:

learning an initial concept;

determining if acceptance criteria are met given the event data;

querying historical event data for similar event patterns; and

allowing the user to edit the initial concept based on the historical event data query

19. The article of claim 14, wherein the automatic predicate learning step utilizes one or more abstraction hierarchies.

20. The article of claim 19, wherein the one or more abstraction hierarchies comprise a hierarchy for at least one of a host and an event type.

EVIDENCE APPENDIX

Submitted herewith are:

Exhibit A: Declaration of Prior Invention under 37 C.F.R. §1.131 (signed by inventors Hellerstein and Ma) and accompanying Exhibits 1-3, originally submitted on September 23, 2004

Exhibit B: Declaration of Prior Invention under 37 C.F.R. §1.131 (signed by assignee Anderson) and accompanying Exhibits 1-3, originally submitted on September 20, 2005

Exhibit C: Attorney Affidavit and accompanying Exhibits 1-3, originally submitted on September 20, 2005

EXHIBIT A

IBM Docket No.: YOR920000581US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	J.L. Hellerstein et al.
Docket No.:	YOR920000581US1
Serial No.:	09/731,937
Filing Date:	December 7, 2000
Group:	2144
Examiner:	Thanh T. Nguyen
Title:	Method and System for Machine-Aided Rule Construction for Event Management

---

DECLARATION OF PRIOR INVENTION UNDER 37 C.F.R. §1.131

We, the undersigned, hereby declare and state as follows:

1. We are the named inventors on the above-referenced U.S. patent application.

2. We conceived the invention that is the subject matter of one or more claims of the above-referenced U.S. patent application at least as early as August 2000. At least as early as August 2000, we prepared an internal IBM invention disclosure document entitled "Method and System for Machine-Aided Rule Construction for Event Management." A copy of the above-mentioned document is attached hereto as Exhibit 1.

3. Applicants' attorney, Mr. William E. Lewis, received the above-mentioned document on or about August 28, 2000 with instructions to prepare and file a U.S. patent application based on the above-mentioned document.

4. Applicants' attorney, Mr. William E. Lewis, sent a first draft of the above-referenced U.S. patent application to the inventors on November 27, 2000 under cover of a facsimile cover sheet. A copy of the facsimile cover sheet is attached hereto as Exhibit 2.

5. Applicants' attorney, Mr. William E. Lewis, sent a revised draft of the above-referenced U.S. patent application to the inventors on December 4, 2000 under cover of a facsimile cover sheet. A copy of the facsimile cover sheet is attached hereto as Exhibit 3.

6. The invention was reduced to practice by filing the above-referenced patent application on December 7, 2000.

7. All statements made herein of our own knowledge are true, and all statements made on information and belief are believed to be true.

8. We understand that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. §1001, and may jeopardize the validity of the application or any patent issuing thereon.

Date: 8/31/04

Joseph L. Hellerstein

Date: 08/31/04

Sheng Ma

Date: ____________

David A. Rabenhorst

EXHIBIT 1

Method and System for Machine-Aided Rule Construction for Event Management

U.S. PATENT DOCUMENTS

•5,874,955	3/12/96	Rogowitz, Rabenhorst, Treinish
•5,661,668	8/26/97	Yemini; Yechiam ,Yemini; Shaula, Kliger; Shmuel ("Apparatus and method for analyzing and correlating events in a system using a causality matrix")

OTHER PUBLICATIONS

- Computer Associates International, "Neugents. The Software that can Think," July 16, 1999, http://www.cai.com/neugents.
- Sheng Ma and Joseph L. Hellerstein, ``EventBrowser: A Flexible Tool for Scalable Analysis of Event Data," Distributed Operations and Management, 1999.
- K.R. Milliken et al. "YES/MVS and the automation of operations for large computer complexes," IBM Systems Journal, Vol 25, No. 2, 1986.
- Tom M. Mitchell. **Machine Learning**. McGraw Hill, 1997.

ABSTRACT

Methods and systems are described for learning correlation rules used in event management. The key method consists of the steps of (a) marking one or more event groupings; (b) employing a machine learning program to learn the underlying concept of these groupings; (c) including a rule right-hand side; and (d) putting the new rule in the Rule DB. The system to implement this method consists of components for: (1) interactive visualization and user interface control; (2) query-based learning; (3) Event DB access; and (4) correlation Rule DB access.

FIELD OF THE INVENTION

The present invention relates generally to network and systems management and more specifically to detecting and resolving availability and performance problems.

BACKGROUND

With the dramatic decline in the price of hardware and software, the cost of ownership for computing devices is increasingly dominated by network and systems management. Included here are tasks such as establishing configurations, help desk support, distributing software, and ensuring the availability and performance of vital services. The latter is particularly important since inaccessible and/or slow services decrease revenues and degrade productivity.

The first step in managing availability and performance is event management. Almost all computing devices have a capability whereby the onset of an exceptional condition results in the generation of a message so that potential problems are detected before they lead to widespread service degradation. Such exceptional conditions are referred to as **events**. Examples of events include: unreachable destinations, excessive CPU consumption, and duplicate IP addresses. An event message contains multiple attributes, especially: (a) the source of the event, (b) type of event, and (c) the time at which the event was generated.

Event messages are sent to an **event management system (EMS)**. An EMS has an **adaptor** that parses the event message and translate it into a normalized form (an adaptor). This normalized information is then placed into an **Event DB**. Next, the normalized event is fed into a **correlation engine** that that determines actions to be taken. This determination is typically driven by correlation rules that are kept in a **Rule DB**. Examples of processing done by correlation rules includes:

1. Elimination of duplicate messages. Duplicate is interpreted broadly here. For example, if multiple hosts on the same local area network generate a destination-unreachable message for the same destination, then the events contain the same information.

2. Maintenance of operational state. State may be as simple as which devices are up and which are down. It may be more complex as well, especially for devices that have many intermediate states or special kinds of error conditions (e.g., printers)

3. Problem detection. A problem is present if one or more components of the system are not functioning properly. For example, the controller in a load balancing system may fail in a way so that new requests are always routed to the same back-end web server, a situation that can be tolerated at low loads but can lead to service degradation at high load. Providing early detection of such situations is important in order to ensure that problems do not lead to wide-spread service disruptions.

4. Problem isolation. This involves determining the components that are causing the problem. For example, distributing a new release of an application that has software errors can result in problems for all end-users connecting to servers with the updated application. Other examples of problem causes include: device failure, exceeding some internal limit (e.g., buffer capacity), and excessive resource demands.

The correlation engine provides automation that is essential for delivering cost effective management of complex computing environments. Current art provides three kinds of correlation. The first employs operational policies expressed as rules (e.g., Milliken, 1986). Rules are if-then statements in which the if-part tests the values of attributes of individual event, and the then-part specifies actions to take. An example of such a rule is "If a hub generates an excessive number of interface-down events, then check if the software loaded on the hub is compatible with its hardware release." The industry experience has been that such rules are difficult to construct, especially if they include installation-specific information.

Another approach has been developed by SMARTS (Yemini et al.) based on the concept of a codebook that matches a repertoire of known problems with event sequences observed during operation. Here, operational policies are models of problems and symptoms. Thus, accommodating new problems requires properly modeling their symptoms and incorporating their signatures into the code book. In theory this approach can accommodate installation-specific problems. However, doing so in practice is difficult because of the high level of sophistication required to encode installation-specific knowledge into rules.

Recently, a third approach to event correlation has been proposed (Computer Associates International, 1999). This approach trains a neural network to predict future occurrences of events based on the frequency of their occurrence in historical data. Typically, events are specified based on thresholds, such as CPU utilization exceeding 90%. The policy execution system uses the neural network to determine the likelihood of one of the previously specified events occurring at some time in the future. While this technique can provide advanced knowledge of the occurrence of an event, it still requires

specifying the events themselves. At a minimum, such a specification requires detailing the following:

1. The variable measured (e.g., CPU utilization)

2. The directional change that considered (e.g., too large)

3. The threshold value (e.g., 90%)

The last item can be obtained automatically from examining representative historical data. Further, graphical user interfaces can provide a means to input the information in items (2) and (3). However, it is often very difficult for installations to choose which variables should be measured and the directional change that constitutes an exceptional situation.

To summarize, current art for event manage systems is of three types. The first (e.g., as in Milliken, 1986) requires that correlation rules be specified by experts, a process that is time-consuming and expensive. The second (e.g., as in Yemini) reduces the involvement of experts but only for aspects of event management that share broad commonalties (e.g., IP connectivity). The third (e.g., Computer Associates International, 1999) attempts to automate the construction of correlation rules for a broader range of management areas. However, to date, this has not been done in a manner that provides for customization by experts, especially in a way that avoids dealing with low-level details (e.g., specific threshold values, the choice of measurement values, and directional changes of interest for these variables).

More broadly, no existing EMS provides decision support for constructing correlation rules, which we refer to as **machine-aided rule construction**. At best, existing art provides authoring systems that aid in syntax checking and formatting. However, no assistance is provided for translating examples of event patterns (drawn from historical data) into correlation rules.

SUMMARY OF THE INVENTION

The present invention addresses the problem of decision support for constructing correlation rules for event management. The invention includes

systems and methods for visualizing event data and machine-based processing of these data to aid in rule construction.

Providing decision support for rule construction requires capabilities for visualizing and describing patterns in terms of rule left-hand sides. In the area of visualization, our starting point is the work described in Ma and Hellerstein, 1999. Also relevant here is Rogowitz, Rabenhorst, and Treinish, 1999 which describes a way to provide preferred visualizations.

Our invention makes use of machine learning algorithms to describe patterns in terms of rules. The framework we adopt is learning concepts expressed as predicates on attributes (e.g., as in Mitchell, 1997). In essence, a concept is a where-clause as expressed in the structured query language (SQL). An example is

> *All events originate from subnet 15.2.3 and the* event rate exceeds .75 per second.

Here, the attribute subnet must have the value 15.2.3 and the total number of events divided by the time-span in seconds of the group must exceed .75.

Learning concepts is greatly facilitated by using one or more **abstraction hierarchy**. An example of a two level hierarchy is: (a) the host itself (e.g., yahoo.com) and (b) its type (e.g., file server, web server, name server). A more extensive hierarchy might be based on the kind of interactions (workload) being done such as: (i) the specific transaction, (ii) the user performing the transaction, (iii) the user's department, and (iv) the division in which the user works. If both cases, we have containment in that higher levels encompass lower ones. In event management, there are often multiple hierarchies (e.g., time, configuration, workload, event type).

A broad class of learning algorithms we consider (e.g., generalization-specialization algorithms as in Mitchell, 1997) uses abstraction hierarchies in two ways. First, when a positive example is encountered that is not covered by the current set of predicates, the level of one or more abstraction hiearchies is increased to include this example. Second, when a negative example is encountered that is covered by the predicate, the level of one or more abstraction hierarchies is decreased. Various schemes are used to optimize that hierarchy levels chosen to maximize the number of positive examples covered and minimize the number of negative examples covered..

The main method of our invention consists of a series of interactions between an analyst--a domain expert--and our system (hereafter, referred to as the machine) whereby correlation rules are constructed. This method consists of steps for (1) the analyst marking one or more event groupings; (2) the machine learning the left-hand side for event patterns; (3) the analyst adding the right-hand side; and (4) the machine placing the rule in the Rule DB that is used by the correlation engine of the EMS. Step (1) is greatly aided by having an effective tool for visualizing and interacting with event groups. Step (2) employs machine-learning techniques, especially query-based learning and generalization-specialization hierarchies that allow a machine to choose the best level of an abstraction hierarchy to cover positive examples of an event pattern (and avoid negative examples).

To elaborate, in our invention learning a left-hand side means determining the predicates necessary to describe a set of event groupings. Predicates consist of logical statements about attribute values. For example, it may be that event groups are characterized originating from hubs, on subnet 9.2.16, with an event rate of .5/second. Then, we want a learning algorithm to determine these predicates.

The foregoing method may be augmented by incorporating data mining algorithms to aid in finding patterns of interest. These patterns can, in essence, seed the process of finding left-hand sides of rules.

The system of our invention includes components for interactive visualization, learning event patterns, rule construction, event data access, and Rule DB access. The visualization system in conjunction with event data access provide a means for analysts to select event groupings that are then translated into left-hand sides by the pattern learner. Rule construction in combination with Rule DB access provide a means for adding the rule right-hand side and placing the result in the Rule DB.

Considerable benefits accrue from our invention. First, the construction of correlation rules is made easier in that left-hand sides of rules can be generated automatically. Clearly, this is a productivity benefit in that expressing left-hand sides can be time consuming. In addition, this capability can allow those who are knowledgeable about operations to develop rules even though they may not be trained in constructing correlation rules.

Another benefit of the present invention relates to the assessment of correlation rules once they are constructed. In existing art, rules are evaluated by using them in production systems. While in some sense this is the ultimate test, it may be some time before a situation arise when the rule is invoked. A

complementary approach is to apply the rule's left-hand side to historical data, selecting instances of the patterns specified by the rule. By so doing, the operations staff can determine if the situations for which the rule is intended are in fact those that will be selected in production.

BRIEF DESCRIPTION OF DRAWINGS

Fig. 1 shows the overall architecture in which our invention operates.

Fig. 2 displays a visualization used to identify groupings of events when learning the left-hand side of rules.

Fig. 3 shows the process for machine-aided rule construction.

Fig. 4 describes the process for query-based learning of a rule left-hand side.

Fig. 5 displays some hierarchies used in employing the generalization-specialization algorithm to learn rule left-hand sides.

Fig. 6 illustrates the system for machine-aided rule construction.

Fig. 7 shows the system in our invention for pattern learning

Fig. 8 displays a user interface for inputting the information in box 500 in Fig. 4.

Fig. 9 displays a user interface for presenting the results of box 520 in Fig. 4.

Fig. 10 displays a user interface for accomplishing box 530 of Fig. 4.

EMBODIMENT

Fig. 1 displays the overall architecture of the environment in which our invention operates. An operator (100) receives alerts and initiates responding actions based on interactions with the event management system (110) that receives events (170) from various devices, such as file servers (140), name servers (150), and mail servers (160). The event management system updates

the Event DB (180) with newly received events and reads this DB to do event correlation based on the Rule DB (185). An analyst (120) uses the event management decision support system (130) of our invention to develop the correlation rules used by the event management system to control the interactions with the operator. Doing so requires reading historical event data in the Event DB and writing the Rule DB.

Fig. 2 illustrates the kind of display used by the event management decision support system to identify patterns that should be translated into the left-hand side of rules. Many patterns may be present, including periodicities and event bursts. Also, patterns may exist at multiple time-scales.

Fig. 3 displays the process for machine-aided rule construction in our invention. Boxes beginning with an "A" are performed by an analyst (human); those that begin with an "M" are done by the machine; and those with "A,M" are done collaboratively by the analyst and the machine. In 410, the analyst and the machine collaborate to learn the left-hand side of rules based on patterns identified in visualizations such as those displayed in Fig. 2. In 420, the analyst augments the left-hand side with a right-side action. In 430, the machine places the new rule in the Rule DB of the event management system (which is box 330 in Fig. 6).

Fig. 4 provides the details of step 410 in Fig. 3. In 500, the analyst marks one or more event groupings and indicates if they are positive or negative examples of the problem to be detected in the rule's left-hand side. In 505, the machine learns a concept using the positive and negative examples provided by the analyst. In 510 the machine determines if there are a sufficient number of examples to learn the rule left-hand side. If there are, the flow proceeds to step 420. If there is not, the machine looks for similar patterns based on the rule constructed so far. In 530, the analyst critiques the result by determining if the examples to date accurately reflect the concept to be identified. This may involve: (a) reclassifying a positive example as a negative example or a negative example as a positive example; (b) deleting examples; and (c) including or excluding events in an example so that it better conforms with the concept being learned. In 540, the analyst may optionally adjust the parameters of the learner to better operate with the concept being learned. Then, the flow continues with box 505.

To elaborate on step 520, consider the preliminary concept "there is a port-down event followed by a port-up event from the same host in within 5 seconds". The machine seeks other examples of such an event sequence from a single host. One way this can be done is for the machine to do a SQL query that retrieves all event interface-down events. Then for each, the machine also retrieves the events that occurred over the next five seconds from that same

host. The machine then checks if one of these events is an interface-up. For those hosts that this is the case, the machine then reports the entire sequence of events from interface-down through interface-up.

Fig. 5 displays examples of the hierarchies used to learn concepts for the left-hand side of rules. Four hierarchies are shown. In 600, there is a time hierarchy consisting of work shift (610), hour of the day (625), and minute within the hour (630). In 605, there is a configuration element hierarchy consisting of the type of host (635) (e.g., mail server, file server) and the host identifier (640) (e.g., its Internet address). In 610, there is a workload hierarchy consisting of the division in which the user is employed (645), the department (650), and user's name (655), and the transaction (or work unit) being performed by the user (660). In 615, there is a hierarchy of event types consisting of the situation in which the event occurs (e.g., printer failure) (665), the nature of the action (670), and the specifics of the element itself (675).

Fig. 6 displays the components of the event management decision support system (130). The interactive visualization and user interface control (700) provides overall control of the interactions with the analyst (120) and the flow within the event management decision support system. The pattern learner (710) is invoked to perform step 410 described in Fig. 3, and the rule constructor (730) in collaboration with the Rule DB access component (725) are used to perform step 430 in Fig. 3, which involves reading and writing the Rule DB. The event miner (705) and event data access (720) components are used in combination to aid in visualization and similarity query functionality used in step 520 of Fig. 4, which requires reading the Event DB.

Fig. 7 details the elements of the pattern learner (710) in Fig. 6. The event visualization and control component (800) controls interactions with the analyst for purposes of learning event patterns. 800 also controls the flow within the pattern learner, including queries to the Event DB via the constraint query engine (805), which in turn invokes the Event Data Access component (810) to read from the Event DB. In addition, 800 invokes the Pattern Inference component (815) to determine possible patterns in the set of positive and negative examples, and establishes hierarchies used by the Hierarchy Manipulator (825) that is employed by 815. Event Visualization and Control also updates the set of positive and negative examples (820) and invokes the Similarity Query Engine (830) to aid in finding other positive and negative examples. Doing so requires specifying numerical distances between patterns, which 800 specifies through interactions with the Distance Calculator (835), a component that is invoked by the Similarity Query Engine.

Fig. 8, 9, and 10 illustrate how analysts interact with the system in our invention to construct correlation rules. Fig. 8 illustrates a user interface that can

be employed to achieve box 500 in Fig. 4. Seen here is a scatter plot. The x-axis is time and the y-axis is the host from which an event originated. The latter is categorical and so can be ordered in any manner that is convenient for visualizing patterns. The user has selected a set of events between times 5 and 9 and for hosts C, B, and A. This is an example of an underlying concept that the analyst knows but cannot articulate in a computer encoded manner. For the purposes of this example, the true concept contains all events that are on subnet 15.2.3 and for which the rate of the event set (count divided by time) exceeds .5/second.

Fig. 9 displays the results of steps 505, 510, and 520 in Fig. 4. Here, the machine has inferred from the example of Fig. 8 the concept that events originate from a Hub and that the rate exceeds .75/second. We see two new patterns that have been found by the machine that are consistent with this concept.

Fig. 10 shows how the analyst accomplishes step 530 in Fig. 4. The analyst has edited the patterns found by the machine to be consistent with the analyst's concept. Specifically, in one case an event is excluded; in the other, a previously excluded event is included.

CLAIMS

1. System for containing an event visualization, data access component, event group selection and editing, a similarity query, and a concept learner component to provides for learning situations.

2. Method for the system in claim (1) for learning situations that includes the steps of marking event groups as positive and negative examples, (2) inferring concepts from the examples, and (3) translating this into the left-hand side of a rule.

3. A method as in claim 2 in which the learning step includes: (a) learning an initial concept, (b) determining if this is statistically significant given the data, (c) querying historical event data for similar event groupings, and (d) allowing the user to critique the result.

4. Concept hierarchy used by the method in claim (2) that includes hierarchies for hosts and event types.

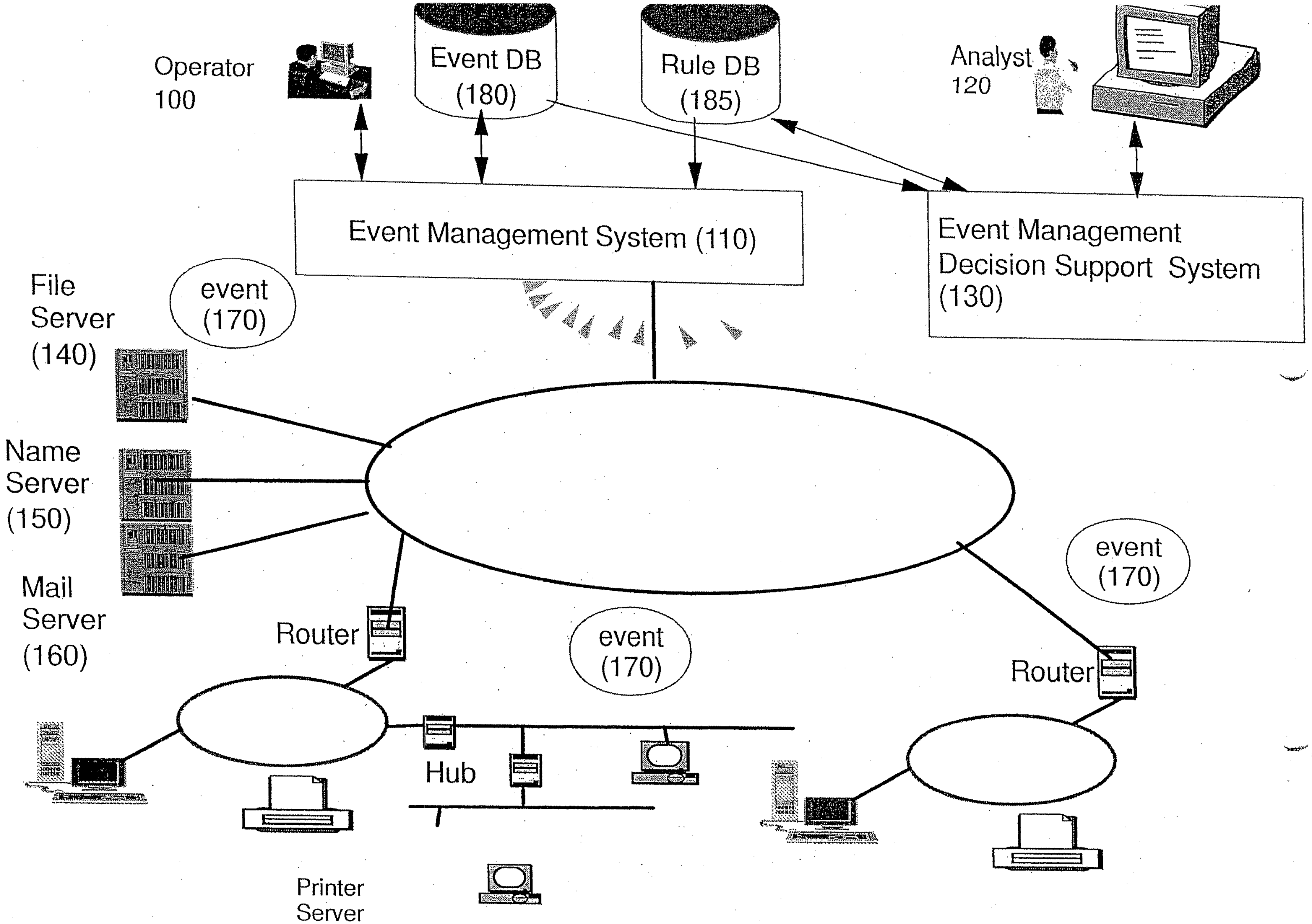

Operator
100
Event DB
(180)
Rule DB
(185)
Analyst
120
Event Management System (110)
Event Management
Decision Support System
(130)
File
Server
(140)
event
(170)
Name
Server
(150)
Mail
Server
(160)
Router
event
(170)
event
(170)
Router
Hub
Printer
Server

Fig. 1

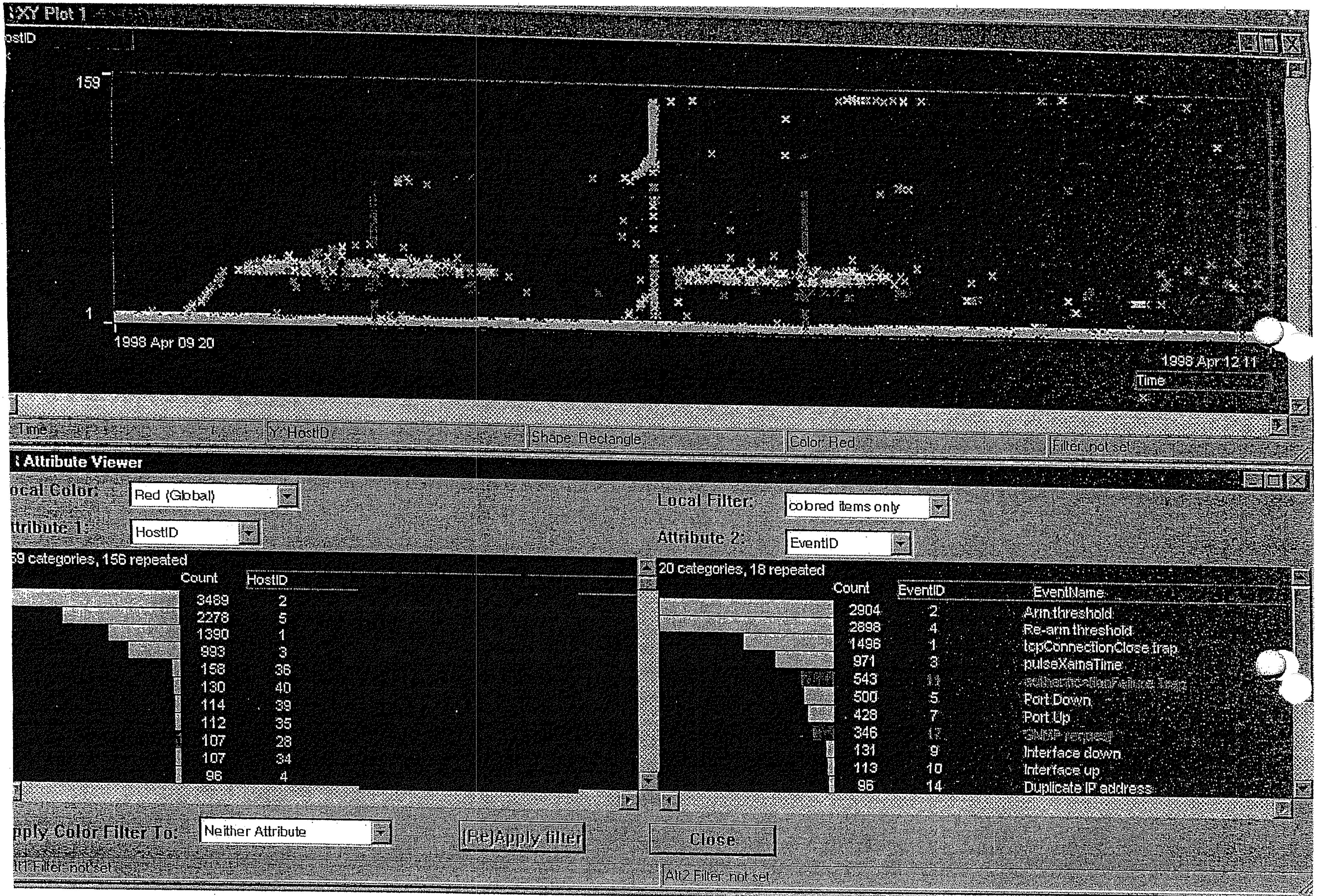

XY Plot 1
ostID
159
1
1998 Apr 09 20
1998 Apr 12 11
Time
Y: HostID
Shape: Rectangle
Color: Red
Filter: not set
Attribute Viewer
ocal Color:
Red (Global)
ttribute 1:
HostID
59 categories, 156 repeated
Count
HostID
3489 2
2278 5
1390 1
993 3
158 36
130 40
114 39
112 35
107 28
107 34
96 4
Local Filter:
colored items only
Attribute 2:
EventID
20 categories, 18 repeated
Count
EventID
EventName
2904 2 Arm threshold
2898 4 Re-arm threshold
1496 1 tcpConnectionClose trap
971 3 pulseXamsTime
543
500 5 Port Down
428 7 Port Up
346
131 9 Interface down
113 10 Interface up
96 14 Duplicate IP address
pply Color Filter To:
Neither Attribute
(Re)Apply filter
Close
tt1 Filter: not set
Att2 Filter: not set

Fig. 2

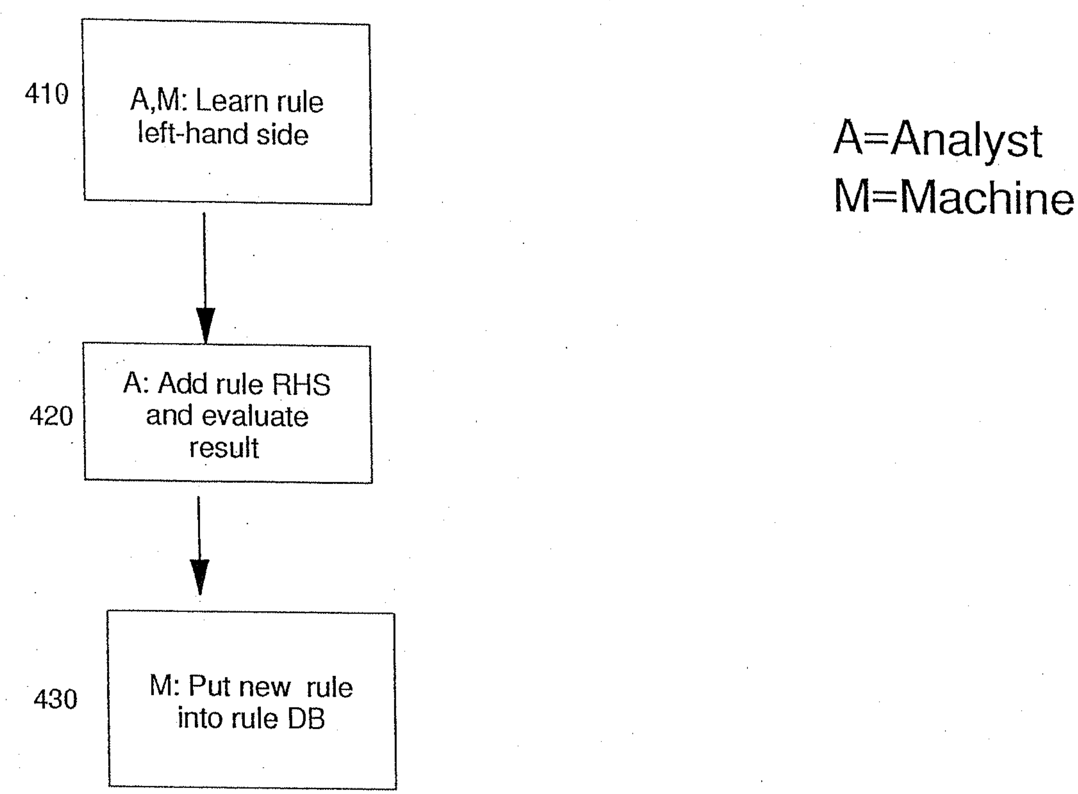

410
A,M: Learn rule
left-hand side
420
A: Add rule RHS
and evaluate
result
430
M: Put new rule
into rule DB
A=Analyst
M=Machine

Fig. 3

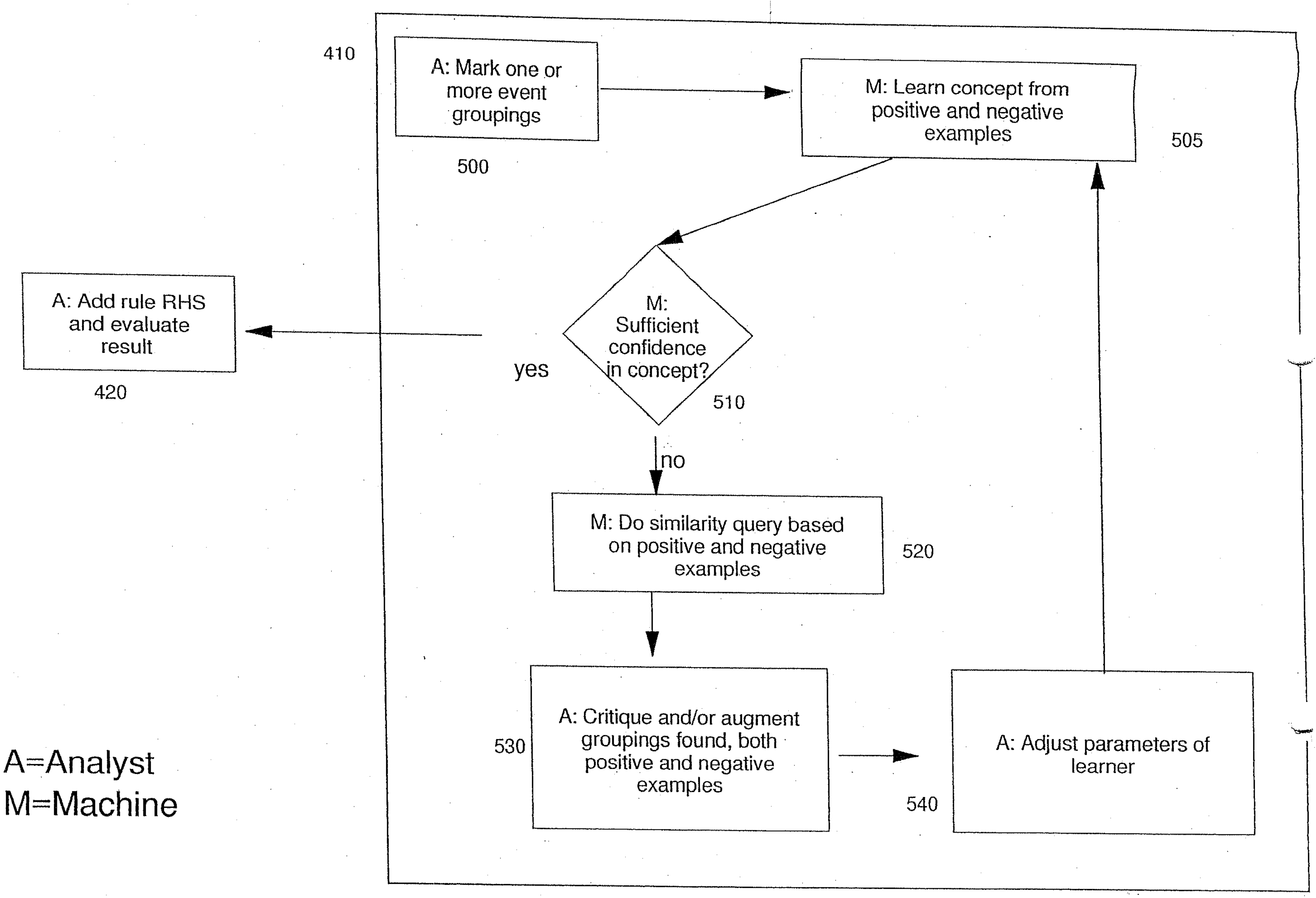

410
A: Mark one or more event groupings
500
M: Learn concept from positive and negative examples
505
M: Sufficient confidence in concept?
510
yes
no
A: Add rule RHS and evaluate result
420
M: Do similarity query based on positive and negative examples
520
A: Critique and/or augment groupings found, both positive and negative examples
530
A: Adjust parameters of learner
540
A=Analyst
M=Machine

Fig. 4

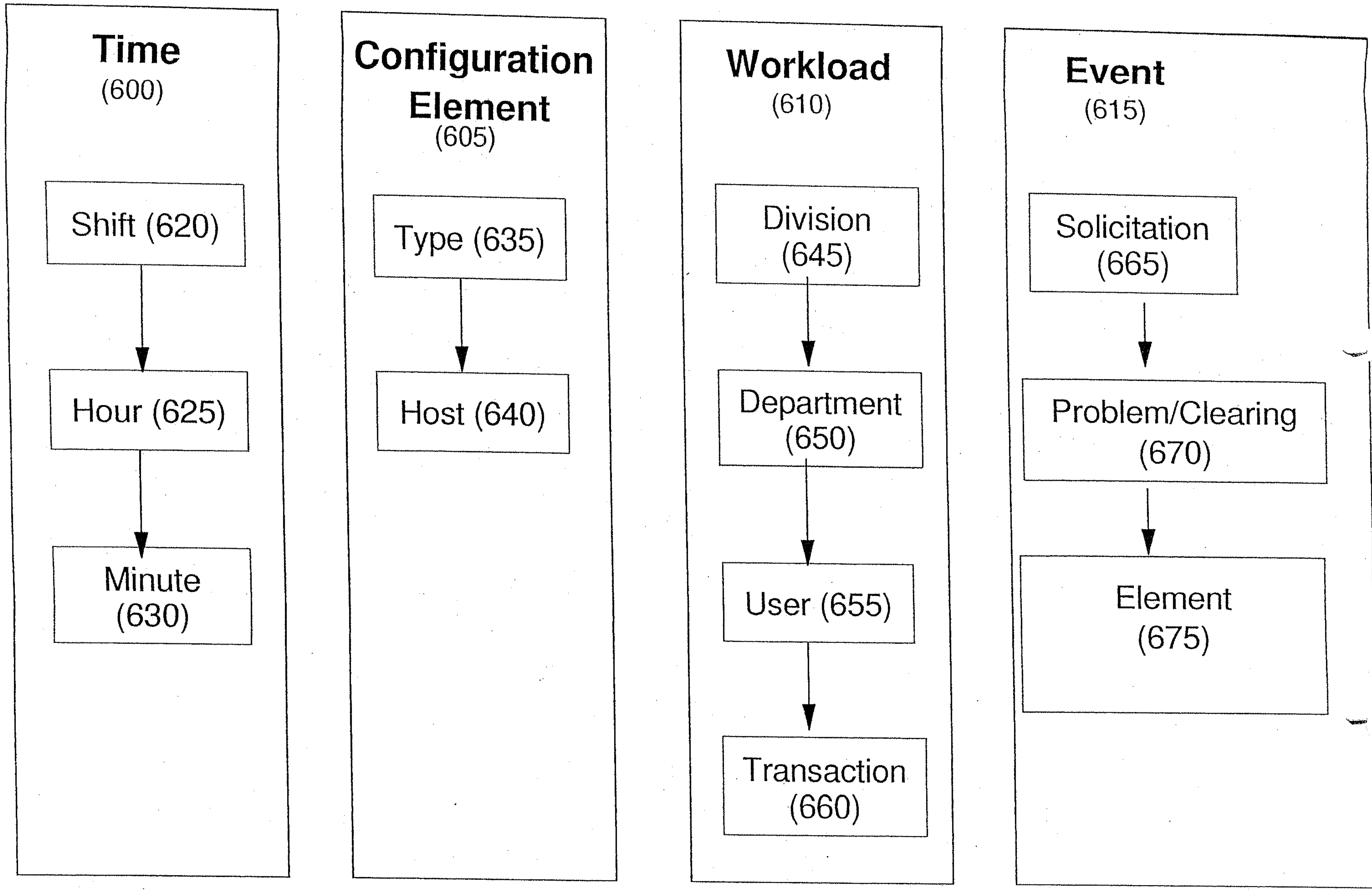

Time
(600)
Shift (620)
Hour (625)
Minute (630)
Configuration Element
(605)
Type (635)
Host (640)
Workload
(610)
Division (645)
Department (650)
User (655)
Transaction (660)
Event
(615)
Solicitation (665)
Problem/Clearing (670)
Element (675)

Fig. 5

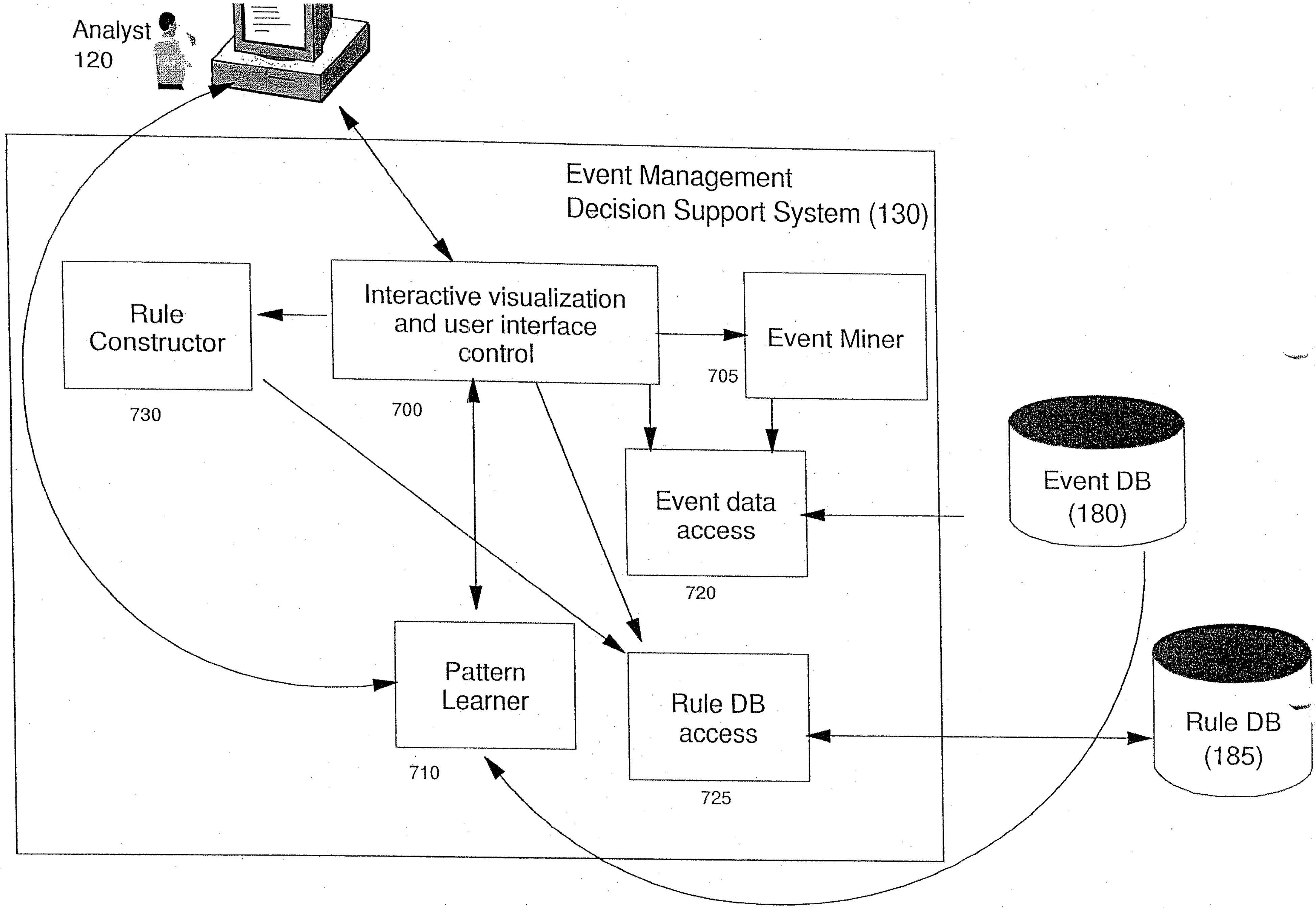
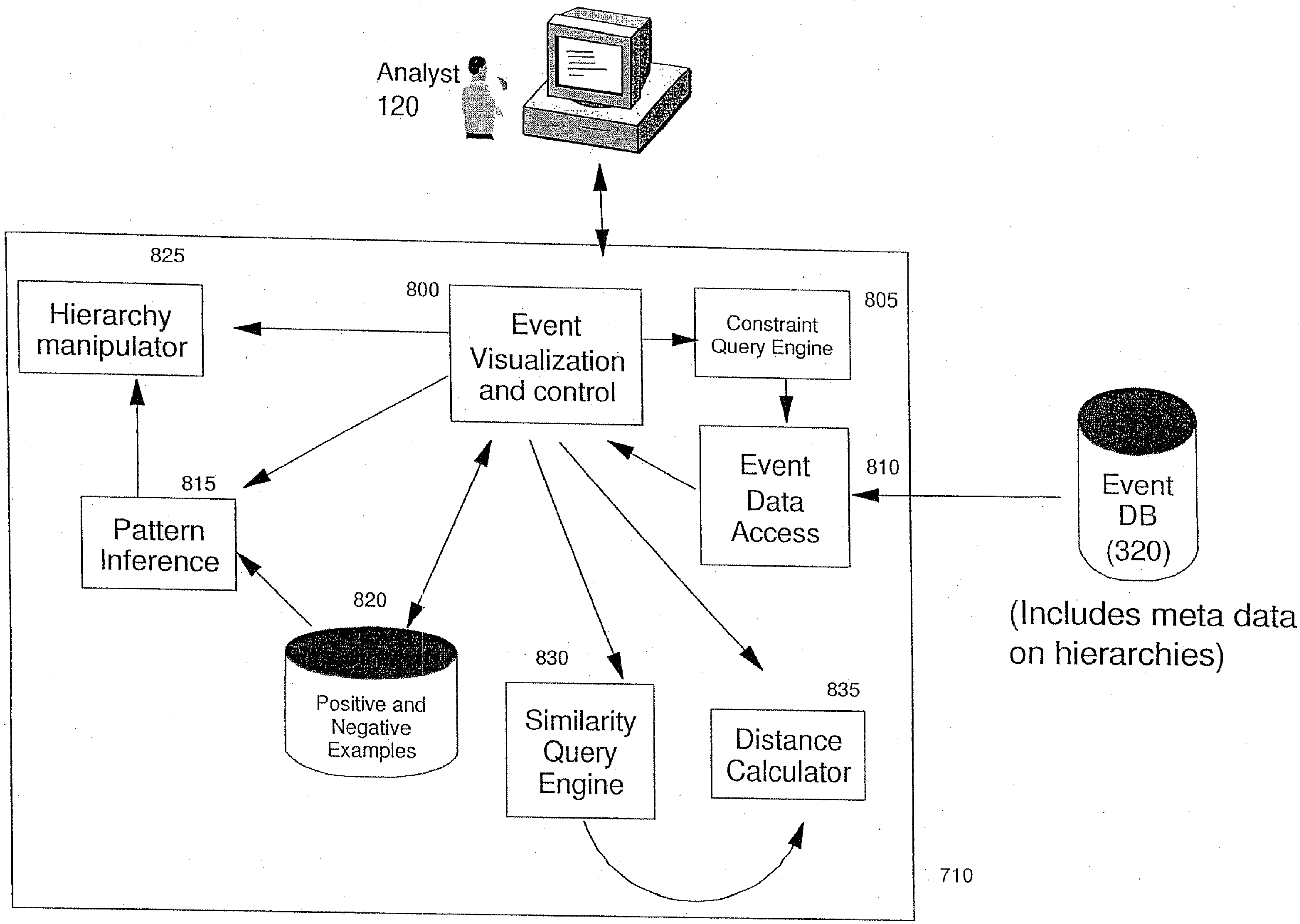
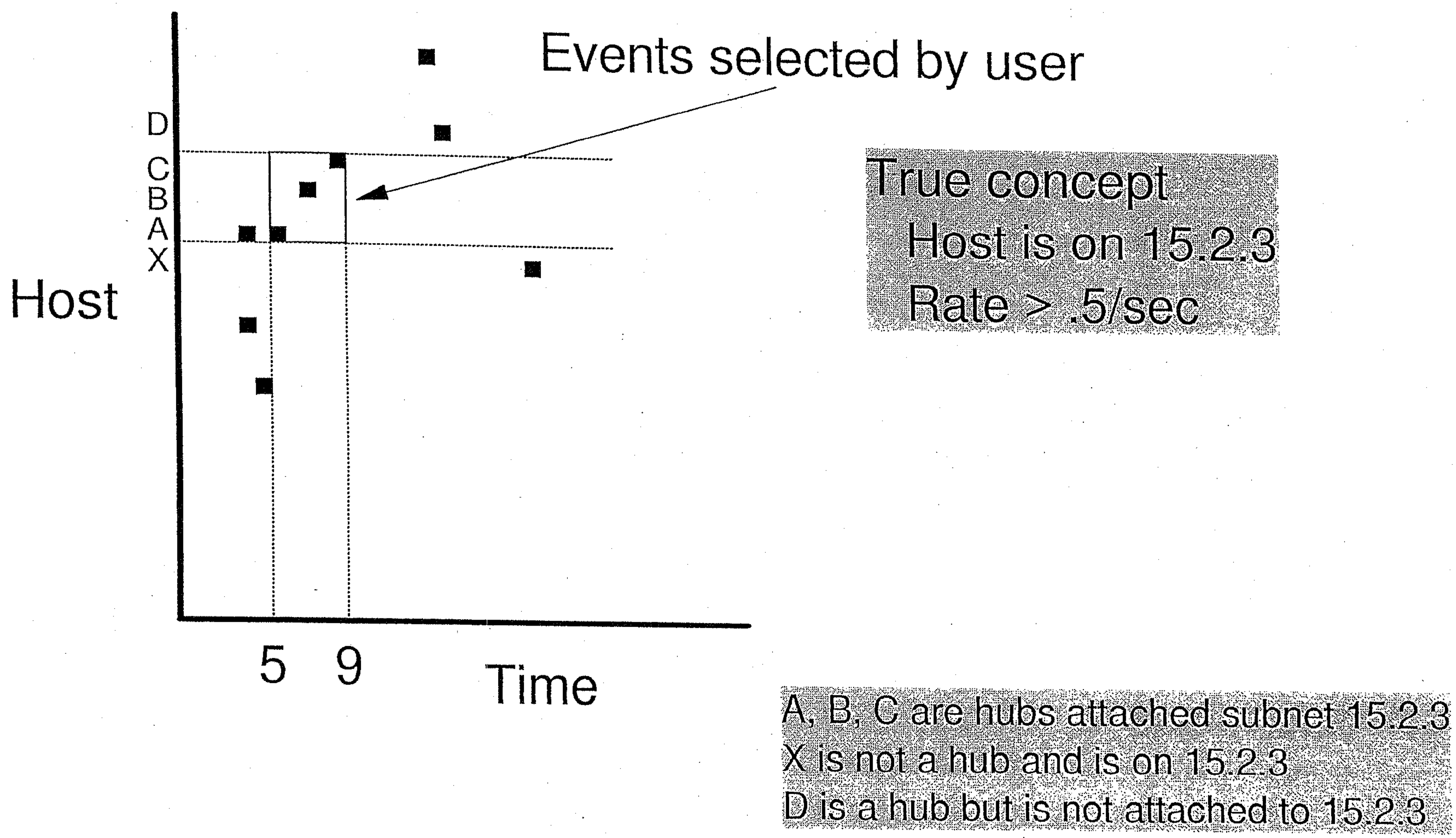

Analyst
120
Event Management
Decision Support System (130)
Rule
Constructor
730
Interactive visualization
and user interface
control
700
Event Miner
705
Event data
access
720
Pattern
Learner
710
Rule DB
access
725
Event DB
(180)
Rule DB
(185)

Fig. 6



Analyst
120
825
Hierarchy manipulator
800
Event Visualization and control
Constraint Query Engine
805
810
Event Data Access
Event DB (320)
(Includes meta data on hierarchies)
815
Pattern Inference
820
Positive and Negative Examples
830
Similarity Query Engine
835
Distance Calculator
710

Fig. 7



Events selected by user
D
C
B
A
X
Host
5
9
Time
True concept
Host is on 15.2.3
Rate > .5/sec
A, B, C are hubs attached subnet 15.2.3
X is not a hub and is on 15.2.3
D is a hub but is not attached to 15.2.3

Fig. 8

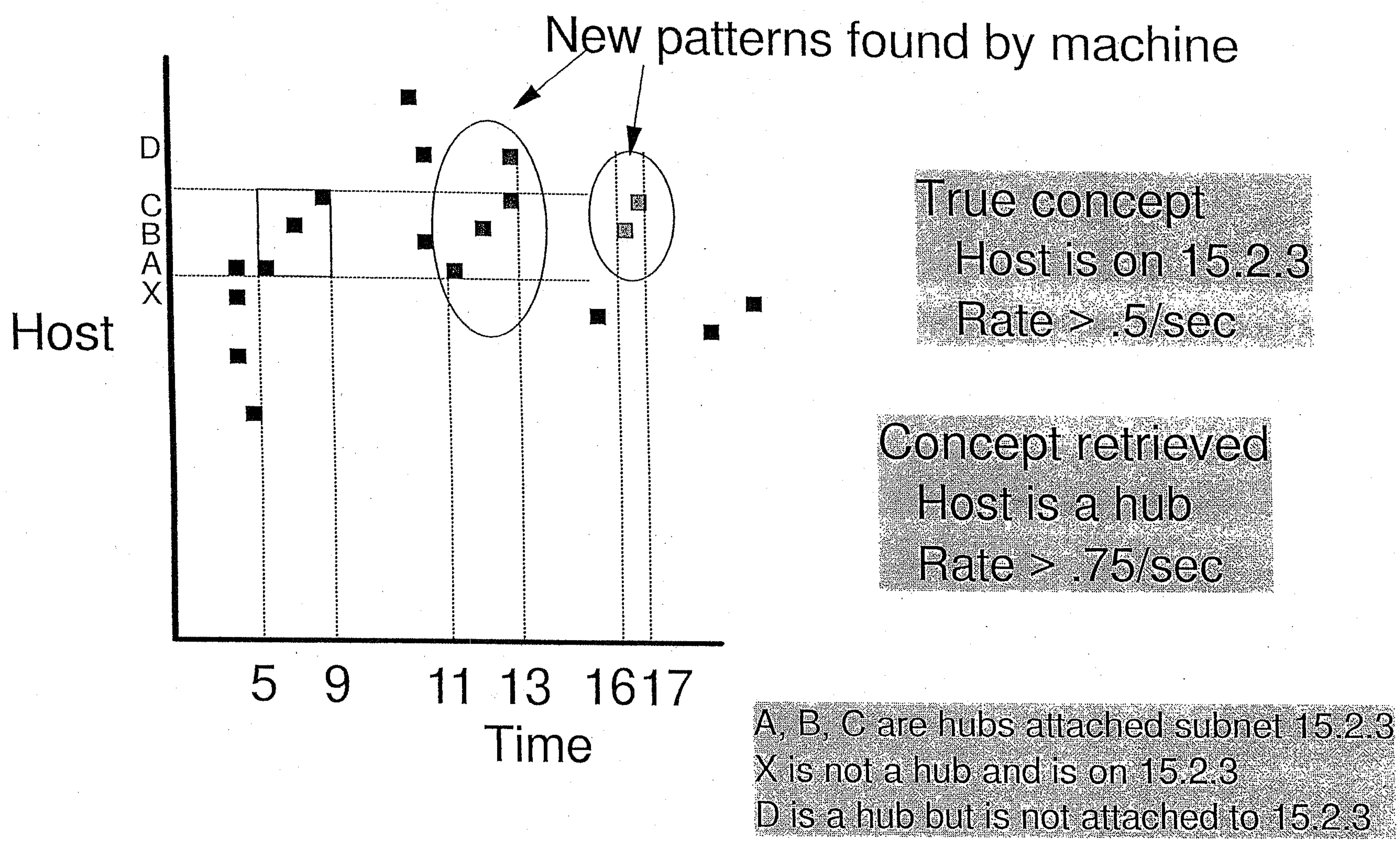

New patterns found by machine
Host
D
C
B
A
X
5
9
11
13
1617
Time
True concept
Host is on 15.2.3
Rate > .5/sec
Concept retrieved
Host is a hub
Rate > .75/sec
A, B, C are hubs attached subnet 15.2.3
X is not a hub and is on 15.2.3
D is a hub but is not attached to 15.2.3

Fig. 9

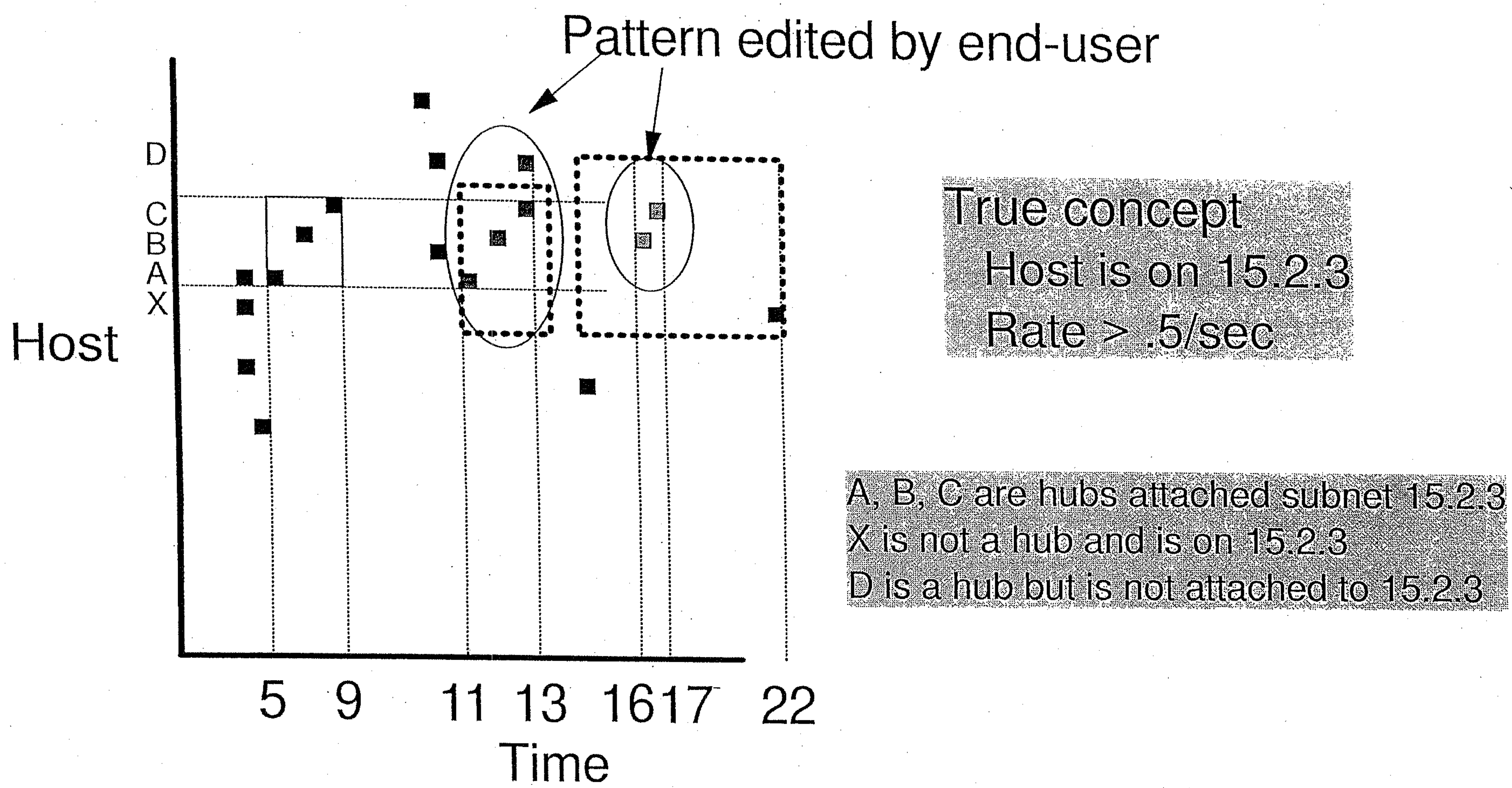

Pattern edited by end-user
Host
D
C
B
A
X
5
9
11
13
1617
22
Time
True concept
Host is on 15.2.3
Rate > .5/sec
A, B, C are hubs attached subnet 15.2.3
X is not a hub and is on 15.2.3
D is a hub but is not attached to 15.2.3

Fig. 10

EXHIBIT 2

RYAN, MASON & LEWIS, LLP
ATTORNEYS AT LAW
90 FOREST AVENUE
LOCUST VALLEY, NEW YORK 11560
Telephone: (516) 759-2946
Facsimile: (516) 759-9512
Email: wel@rml-law.com

DATE: November 27, 2000	**FILE:** YOR920000581US1

Facsimile Message From: **WILLIAM E. LEWIS**

Please deliver the following pages to:

NAME:	Joseph L. Hellerstein
OF:	IBM Corporation
FAX NUMBER:	(914) 784-6040
NUMBER OF PAGES INCLUDING THIS COVER PAGE:	33
COMMENTS/INSTRUCTIONS:	

Dear Joe:

Please find attached a draft of the patent application relating to your invention. I kindly ask that you review the attached draft, and have your co-inventors do the same, and provide me with your comments at your earliest convenience. Please confirm receipt. Thank you for your assistance.

Best regards,

Bill Lewis

Bill Lewis

If you do not receive all of the pages, please call us back as soon as possible at (516) 759-2946.

EXHIBIT 3

RYAN, MASON & LEWIS, LLP
ATTORNEYS AT LAW
90 FOREST AVENUE
LOCUST VALLEY, NEW YORK 11560
Telephone: (516) 759-2946
Facsimile: (516) 759-9512
Email: wel@rml-law.com

DATE: December 4, 2000	**FILE:** YOR920000581US1

Facsimile Message From: **WILLIAM E. LEWIS**

Please deliver the following pages to:

NAME: Joseph L. Hellerstein
OF: IBM Corporation
FAX NUMBER: (914) 784-6040
NUMBER OF PAGES INCLUDING THIS COVER PAGE: 36
COMMENTS/INSTRUCTIONS:

Dear Joe:

Please find attached a revised draft of the patent application relating to your invention. The revised draft incorporates your recent comments. I kindly ask that you review the attached revised draft in its entirety, and have your co-inventors do the same, and provide me with your comments as soon as possible. As you know, we plan to file this application on or before **Friday, December 8, 2000**. Please confirm receipt. Thank you again for your assistance.

Best regards,

Bill

Bill Lewis

If you do not receive all of the pages, please call us back as soon as possible at (516) 759-2946.

EXHIBIT B

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s):	J.L. Hellerstein et al.
Case:	YOR920000581US1
Serial No.:	09/731,937
Filing Date:	December 7, 2000
Group:	2144
Examiner:	Thanh T. Nguyen
Title:	Method and System for Machine-Aided Rule Construction for Event Management

---

DECLARATION OF PRIOR INVENTION UNDER 37 C.F.R. §1.131

I, the undersigned, hereby declare and state as follows:

1. I am authorized to act on behalf of the named assignee of the above-referenced U.S. patent application.

2. The named inventors conceived the invention that is the subject matter of one or more claims of the above-referenced U.S. patent application at least as early as August 2000. At least as early as August 2000, the named inventors prepared an internal IBM invention disclosure document entitled "Method and System for Machine-Aided Rule Construction for Event Management." A copy of the above-mentioned document is attached hereto as Exhibit 1.

3. Applicants' attorney, Mr. William E. Lewis of Ryan, Mason & Lewis, LLP, received the above-mentioned document on or about August 28, 2000 with instructions to prepare and file a U.S. patent application based on the above-mentioned document.

4. Applicants' attorney, Mr. William E. Lewis, sent a first draft of the above- referenced U.S. patent application to inventor Joseph L. Hellerstein on November 27, 2000 under cover of a facsimile cover sheet. A copy of the facsimile cover sheet is attached hereto as Exhibit 2.

5. Applicants' attorney, Mr. William E. Lewis, sent a revised draft of the above-referenced U.S. patent application to inventor Joseph L. Hellerstein on December 4, 2000 under cover of a facsimile cover sheet. A copy of the facsimile cover sheet is attached hereto as Exhibit 3.

6. The invention was constructively reduced to practice by filing the above-referenced patent application on December 7, 2000.

7. All statements made herein of my own knowledge are true, and all statements made on information and belief are believed to be true.

8. I understand that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001, and may jeopardize the validity of the application or any patent issuing thereon.

Date: 9-20-05

Lynne D. Anderson

Lynne D. Anderson
Program Manager, Patent Office Liaison

EXHIBIT 1

Method and System for Machine-Aided Rule Construction for Event Management

U.S. PATENT DOCUMENTS

•5,874,955 3/12/96 Rogowitz, Rabenhorst, Treinish

•5,661,668 8/26/97 Yemini; Yechiam ,Yemini; Shaula, Kliger; Shmuel ("Apparatus and method for analyzing and correlating events in a system using a causality matrix")

OTHER PUBLICATIONS

- Computer Associates International, "Neugents. The Software that can Think," July 16, 1999, http://www.cai.com/neugents.
- Sheng Ma and Joseph L. Hellerstein, ``EventBrowser: A Flexible Tool for Scalable Analysis of Event Data," Distributed Operations and Management, 1999.
- K.R. Milliken et al. "YES/MVS and the automation of operations for large computer complexes," IBM Systems Journal, Vol 25, No. 2, 1986.
- Tom M. Mitchell. **Machine Learning**. McGraw Hill, 1997.

ABSTRACT

Methods and systems are described for learning correlation rules used in event management. The key method consists of the steps of (a) marking one or more event groupings; (b) employing a machine learning program to learn the underlying concept of these groupings; (c) including a rule right-hand side; and (d) putting the new rule in the Rule DB. The system to implement this method consists of components for: (1) interactive visualization and user interface control; (2) query-based learning; (3) Event DB access; and (4) correlation Rule DB access.

FIELD OF THE INVENTION

The present invention relates generally to network and systems management and more specifically to detecting and resolving availability and performance problems.

BACKGROUND

With the dramatic decline in the price of hardware and software, the cost of ownership for computing devices is increasingly dominated by network and systems management. Included here are tasks such as establishing configurations, help desk support, distributing software, and ensuring the availability and performance of vital services. The latter is particularly important since inaccessible and/or slow services decrease revenues and degrade productivity.

The first step in managing availability and performance is event management. Almost all computing devices have a capability whereby the onset of an exceptional condition results in the generation of a message so that potential problems are detected before they lead to widespread service degradation. Such exceptional conditions are referred to as **events**. Examples of events include: unreachable destinations, excessive CPU consumption, and duplicate IP addresses. An event message contains multiple attributes, especially: (a) the source of the event, (b) type of event, and (c) the time at which the event was generated.

Event messages are sent to an **event management system (EMS)**. An EMS has an **adaptor** that parses the event message and translate it into a normalized form (an adaptor). This normalized information is then placed into an **Event DB**. Next, the normalized event is fed into a **correlation engine** that that determines actions to be taken. This determination is typically driven by correlation rules that are kept in a **Rule DB**. Examples of processing done by correlation rules includes:

1. Elimination of duplicate messages. Duplicate is interpreted broadly here. For example, if multiple hosts on the same local area network generate a destination-unreachable message for the same destination, then the events contain the same information.

2. Maintenance of operational state. State may be as simple as which devices are up and which are down. It may be more complex as well, especially for devices that have many intermediate states or special kinds of error conditions (e.g., printers)

3. Problem detection. A problem is present if one or more components of the system are not functioning properly. For example, the controller in a load balancing system may fail in a way so that new requests are always routed to the same back-end web server, a situation that can be tolerated at low loads but can lead to service degradation at high load. Providing early detection of such situations is important in order to ensure that problems do not lead to wide-spread service disruptions.

4. Problem isolation. This involves determining the components that are causing the problem. For example, distributing a new release of an application that has software errors can result in problems for all end-users connecting to servers with the updated application. Other examples of problem causes include: device failure, exceeding some internal limit (e.g., buffer capacity), and excessive resource demands.

The correlation engine provides automation that is essential for delivering cost effective management of complex computing environments. Current art provides three kinds of correlation. The first employs operational policies expressed as rules (e.g., Milliken, 1986). Rules are if-then statements in which the if-part tests the values of attributes of individual event, and the then-part specifies actions to take. An example of such a rule is "If a hub generates an excessive number of interface-down events, then check if the software loaded on the hub is compatible with its hardware release." The industry experience has been that such rules are difficult to construct, especially if they include installation-specific information.

Another approach has been developed by SMARTS (Yemini et al.) based on the concept of a codebook that matches a repertoire of known problems with event sequences observed during operation. Here, operational policies are models of problems and symptoms. Thus, accommodating new problems requires properly modeling their symptoms and incorporating their signatures into the code book. In theory this approach can accommodate installation-specific problems. However, doing so in practice is difficult because of the high level of sophistication required to encode installation-specific knowledge into rules.

Recently, a third approach to event correlation has been proposed (Computer Associates International, 1999). This approach trains a neural network to predict future occurrences of events based on the frequency of their occurrence in historical data. Typically, events are specified based on thresholds, such as CPU utilization exceeding 90%. The policy execution system uses the neural network to determine the likelihood of one of the previously specified events occurring at some time in the future. While this technique can provide advanced knowledge of the occurrence of an event, it still requires

specifying the events themselves. At a minimum, such a specification requires detailing the following:

1. The variable measured (e.g., CPU utilization)

2. The directional change that considered (e.g., too large)

3. The threshold value (e.g., 90%)

The last item can be obtained automatically from examining representative historical data. Further, graphical user interfaces can provide a means to input the information in items (2) and (3). However, it is often very difficult for installations to choose which variables should be measured and the directional change that constitutes an exceptional situation.

To summarize, current art for event manage systems is of three types. The first (e.g., as in Milliken, 1986) requires that correlation rules be specified by experts, a process that is time-consuming and expensive. The second (e.g., as in Yemini) reduces the involvement of experts but only for aspects of event management that share broad commonalties (e.g., IP connectivity). The third (e.g., Computer Associates International, 1999) attempts to automate the construction of correlation rules for a broader range of management areas. However, to date, this has not been done in a manner that provides for customization by experts, especially in a way that avoids dealing with low-level details (e.g., specific threshold values, the choice of measurement values, and directional changes of interest for these variables).

More broadly, no existing EMS provides decision support for constructing correlation rules, which we refer to as **machine-aided rule construction**. At best, existing art provides authoring systems that aid in syntax checking and formatting. However, no assistance is provided for translating examples of event patterns (drawn from historical data) into correlation rules.

SUMMARY OF THE INVENTION

The present invention addresses the problem of decision support for constructing correlation rules for event management. The invention includes

systems and methods for visualizing event data and machine-based processing of these data to aid in rule construction.

Providing decision support for rule construction requires capabilities for visualizing and describing patterns in terms of rule left-hand sides. In the area of visualization, our starting point is the work described in Ma and Hellerstein, 1999. Also relevant here is Rogowitz, Rabenhorst, and Treinish, 1999 which describes a way to provide preferred visualizations.

Our invention makes use of machine learning algorithms to describe patterns in terms of rules. The framework we adopt is learning concepts expressed as predicates on attributes (e.g., as in Mitchell, 1997). In essence, a concept is a where-clause as expressed in the structured query language (SQL). An example is

> *All events originate from subnet 15.2.3 and the* event rate exceeds .75 per second.

Here, the attribute subnet must have the value 15.2.3 and the total number of events divided by the time-span in seconds of the group must exceed .75.

Learning concepts is greatly facilitated by using one or more **abstraction hierarchy**. An example of a two level hierarchy is: (a) the host itself (e.g., yahoo.com) and (b) its type (e.g., file server, web server, name server). A more extensive hierarchy might be based on the kind of interactions (workload) being done such as: (i) the specific transaction, (ii) the user performing the transaction, (iii) the user's department, and (iv) the division in which the user works. If both cases, we have containment in that higher levels encompass lower ones. In event management, there are often multiple hierarchies (e.g., time, configuration, workload, event type).

A broad class of learning algorithms we consider (e.g., generalization-specialization algorithms as in Mitchell, 1997) uses abstraction hierarchies in two ways. First, when a positive example is encountered that is not covered by the current set of predicates, the level of one or more abstraction hiearchies is increased to include this example. Second, when a negative example is encountered that is covered by the predicate, the level of one or more abstraction hierarchies is decreased. Various schemes are used to optimize that hierarchy levels chosen to maximize the number of positive examples covered and minimize the number of negative examples covered..

The main method of our invention consists of a series of interactions between an analyst--a domain expert--and our system (hereafter, referred to as the machine) whereby correlation rules are constructed. This method consists of steps for (1) the analyst marking one or more event groupings; (2) the machine learning the left-hand side for event patterns; (3) the analyst adding the right-hand side; and (4) the machine placing the rule in the Rule DB that is used by the correlation engine of the EMS. Step (1) is greatly aided by having an effective tool for visualizing and interacting with event groups. Step (2) employs machine-learning techniques, especially query-based learning and generalization-specialization hierarchies that allow a machine to choose the best level of an abstraction hierarchy to cover positive examples of an event pattern (and avoid negative examples).

To elaborate, in our invention learning a left-hand side means determining the predicates necessary to describe a set of event groupings. Predicates consist of logical statements about attribute values. For example, it may be that event groups are characterized originating from hubs, on subnet 9.2.16, with an event rate of .5/second. Then, we want a learning algorithm to determine these predicates.

The foregoing method may be augmented by incorporating data mining algorithms to aid in finding patterns of interest. These patterns can, in essence, seed the process of finding left-hand sides of rules.

The system of our invention includes components for interactive visualization, learning event patterns, rule construction, event data access, and Rule DB access. The visualization system in conjunction with event data access provide a means for analysts to select event groupings that are then translated into left-hand sides by the pattern learner. Rule construction in combination with Rule DB access provide a means for adding the rule right-hand side and placing the result in the Rule DB.

Considerable benefits accrue from our invention. First, the construction of correlation rules is made easier in that left-hand sides of rules can be generated automatically. Clearly, this is a productivity benefit in that expressing left-hand sides can be time consuming. In addition, this capability can allow those who are knowledgeable about operations to develop rules even though they may not be trained in constructing correlation rules.

Another benefit of the present invention relates to the assessment of correlation rules once they are constructed. In existing art, rules are evaluated by using them in production systems. While in some sense this is the ultimate test, it may be some time before a situation arise when the rule is invoked. A

complementary approach is to apply the rule's left-hand side to historical data, selecting instances of the patterns specified by the rule. By so doing, the operations staff can determine if the situations for which the rule is intended are in fact those that will be selected in production.

BRIEF DESCRIPTION OF DRAWINGS

Fig. 1 shows the overall architecture in which our invention operates.

Fig. 2 displays a visualization used to identify groupings of events when learning the left-hand side of rules.

Fig. 3 shows the process for machine-aided rule construction.

Fig. 4 describes the process for query-based learning of a rule left-hand side.

Fig. 5 displays some hierarchies used in employing the generalization-specialization algorithm to learn rule left-hand sides.

Fig. 6 illustrates the system for machine-aided rule construction.

Fig. 7 shows the system in our invention for pattern learning

Fig. 8 displays a user interface for inputting the information in box 500 in Fig. 4.

Fig. 9 displays a user interface for presenting the results of box 520 in Fig. 4.

Fig. 10 displays a user interface for accomplishing box 530 of Fig. 4.

EMBODIMENT

Fig. 1 displays the overall architecture of the environment in which our invention operates. An operator (100) receives alerts and initiates responding actions based on interactions with the event management system (110) that receives events (170) from various devices, such as file servers (140), name servers (150), and mail servers (160). The event management system updates

the Event DB (180) with newly received events and reads this DB to do event correlation based on the Rule DB (185). An analyst (120) uses the event management decision support system (130) of our invention to develop the correlation rules used by the event management system to control the interactions with the operator. Doing so requires reading historical event data in the Event DB and writing the Rule DB.

Fig. 2 illustrates the kind of display used by the event management decision support system to identify patterns that should be translated into the left-hand side of rules. Many patterns may be present, including periodicities and event bursts. Also, patterns may exist at multiple time-scales.

Fig. 3 displays the process for machine-aided rule construction in our invention. Boxes beginning with an "A" are performed by an analyst (human); those that begin with an "M" are done by the machine; and those with "A,M" are done collaboratively by the analyst and the machine. In 410, the analyst and the machine collaborate to learn the left-hand side of rules based on patterns identified in visualizations such as those displayed in Fig. 2. In 420, the analyst augments the left-hand side with a right-side action. In 430, the machine places the new rule in the Rule DB of the event management system (which is box 330 in Fig. 6).

Fig. 4 provides the details of step 410 in Fig. 3. In 500, the analyst marks one or more event groupings and indicates if they are positive or negative examples of the problem to be detected in the rule's left-hand side. In 505, the machine learns a concept using the positive and negative examples provided by the analyst. In 510 the machine determines if there are a sufficient number of examples to learn the rule left-hand side. If there are, the flow proceeds to step 420. If there is not, the machine looks for similar patterns based on the rule constructed so far. In 530, the analyst critiques the result by determining if the examples to date accurately reflect the concept to be identified. This may involve: (a) reclassifying a positive example as a negative example or a negative example as a positive example; (b) deleting examples; and (c) including or excluding events in an example so that it better conforms with the concept being learned. In 540, the analyst may optionally adjust the parameters of the learner to better operate with the concept being learned. Then, the flow continues with box 505.

To elaborate on step 520, consider the preliminary concept "there is a port-down event followed by a port-up event from the same host in within 5 seconds". The machine seeks other examples of such an event sequence from a single host. One way this can be done is for the machine to do a SQL query that retrieves all event interface-down events. Then for each, the machine also retrieves the events that occurred over the next five seconds from that same

host. The machine then checks if one of these events is an interface-up. For those hosts that this is the case, the machine then reports the entire sequence of events from interface-down through interface-up.

Fig. 5 displays examples of the hierarchies used to learn concepts for the left-hand side of rules. Four hierarchies are shown. In 600, there is a time hierarchy consisting of work shift (610), hour of the day (625), and minute within the hour (630). In 605, there is a configuration element hierarchy consisting of the type of host (635) (e.g., mail server, file server) and the host identifier (640) (e.g., its Internet address). In 610, there is a workload hierarchy consisting of the division in which the user is employed (645), the department (650), and user's name (655), and the transaction (or work unit) being performed by the user (660). In 615, there is a hierarchy of event types consisting of the situation in which the event occurs (e.g., printer failure) (665), the nature of the action (670), and the specifics of the element itself (675).

Fig. 6 displays the components of the event management decision support system (130). The interactive visualization and user interface control (700) provides overall control of the interactions with the analyst (120) and the flow within the event management decision support system. The pattern learner (710) is invoked to perform step 410 described in Fig. 3, and the rule constructor (730) in collaboration with the Rule DB access component (725) are used to perform step 430 in Fig. 3, which involves reading and writing the Rule DB. The event miner (705) and event data access (720) components are used in combination to aid in visualization and similarity query functionality used in step 520 of Fig. 4, which requires reading the Event DB.

Fig. 7 details the elements of the pattern learner (710) in Fig. 6. The event visualization and control component (800) controls interactions with the analyst for purposes of learning event patterns. 800 also controls the flow within the pattern learner, including queries to the Event DB via the constraint query engine (805), which in turn invokes the Event Data Access component (810) to read from the Event DB. In addition, 800 invokes the Pattern Inference component (815) to determine possible patterns in the set of positive and negative examples, and establishes hierarchies used by the Hierarchy Manipulator (825) that is employed by 815. Event Visualization and Control also updates the set of positive and negative examples (820) and invokes the Similarity Query Engine (830) to aid in finding other positive and negative examples. Doing so requires specifying numerical distances between patterns, which 800 specifies through interactions with the Distance Calculator (835), a component that is invoked by the Similarity Query Engine.

Fig. 8, 9, and 10 illustrate how analysts interact with the system in our invention to construct correlation rules. Fig. 8 illustrates a user interface that can

be employed to achieve box 500 in Fig. 4. Seen here is a scatter plot. The x-axis is time and the y-axis is the host from which an event originated. The latter is categorical and so can be ordered in any manner that is convenient for visualizing patterns. The user has selected a set of events between times 5 and 9 and for hosts C, B, and A. This is an example of an underlying concept that the analyst knows but cannot articulate in a computer encoded manner. For the purposes of this example, the true concept contains all events that are on subnet 15.2.3 and for which the rate of the event set (count divided by time) exceeds .5/second.

Fig. 9 displays the results of steps 505, 510, and 520 in Fig. 4. Here, the machine has inferred from the example of Fig. 8 the concept that events originate from a Hub and that the rate exceeds .75/second. We see two new patterns that have been found by the machine that are consistent with this concept.

Fig. 10 shows how the analyst accomplishes step 530 in Fig. 4. The analyst has edited the patterns found by the machine to be consistent with the analyst's concept. Specifically, in one case an event is excluded; in the other, a previously excluded event is included.

CLAIMS

1. System for containing an event visualization, data access component, event group selection and editing, a similarity query, and a concept learner component to provides for learning situations.

2. Method for the system in claim (1) for learning situations that includes the steps of marking event groups as positive and negative examples, (2) inferring concepts from the examples, and (3) translating this into the left-hand side of a rule.

3. A method as in claim 2 in which the learning step includes: (a) learning an initial concept, (b) determining if this is statistically significant given the data, (c) querying historical event data for similar event groupings, and (d) allowing the user to critique the result.

4. Concept hierarchy used by the method in claim (2) that includes hierarchies for hosts and event types.

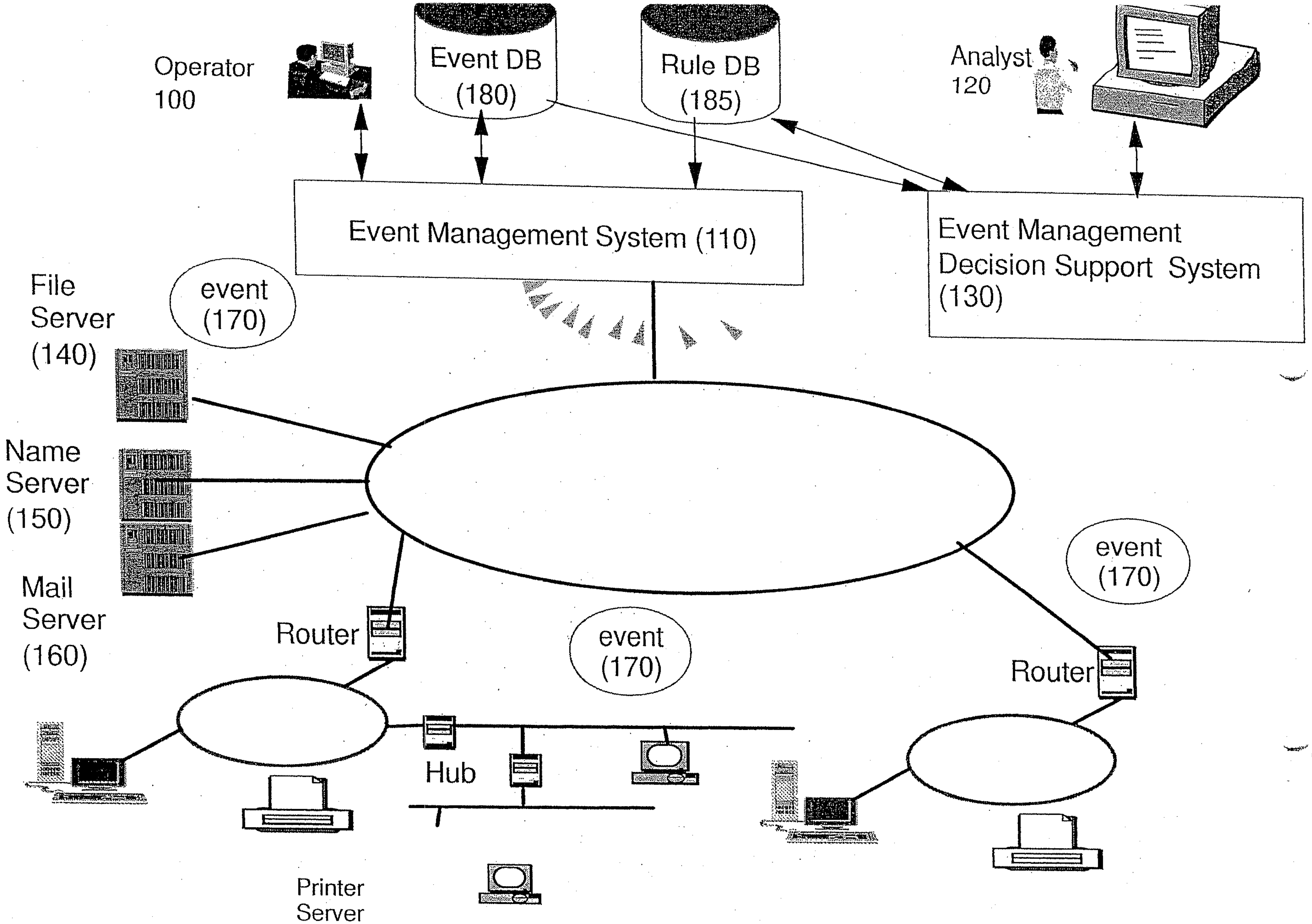

Operator
100
Event DB
(180)
Rule DB
(185)
Analyst
120
Event Management System (110)
Event Management
Decision Support System
(130)
File
Server
(140)
event
(170)
Name
Server
(150)
Mail
Server
(160)
Router
event
(170)
event
(170)
Router
Hub
Printer
Server

Fig. 1

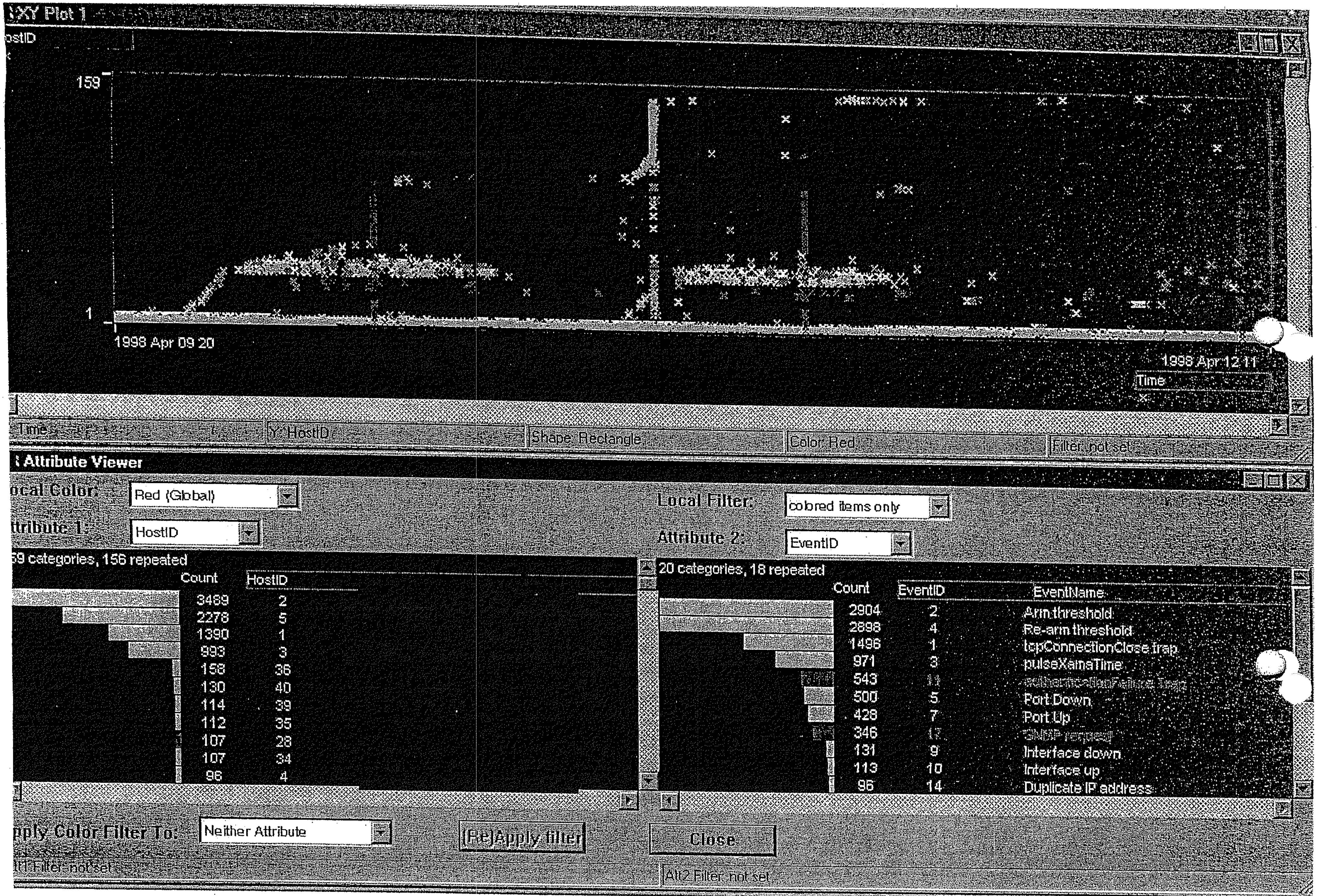

XY Plot 1
ostID
159
1
1998 Apr 09 20
1998 Apr 12 11
Time
Time
Y: HostID
Shape: Rectangle
Color: Red
Filter: not set
Attribute Viewer
ocal Color:
Red (Global)
Local Filter:
colored items only
ttribute 1:
HostID
Attribute 2:
EventID
59 categories, 156 repeated
Count
HostID
3489 2
2278 5
1390 1
993 3
158 36
130 40
114 39
112 35
107 28
107 34
96 4
20 categories, 18 repeated
Count
EventID
EventName
2904 2 Arm threshold
2898 4 Re-arm threshold
1496 1 tcpConnectionClose trap
971 3 pulseXamsTime
543
500 5 Port Down
428 7 Port Up
346
131 9 Interface down
113 10 Interface up
96 14 Duplicate IP address
pply Color Filter To:
Neither Attribute
(Re)Apply filter
Close
tt1 Filter: not set
Att2 Filter: not set

Fig. 2

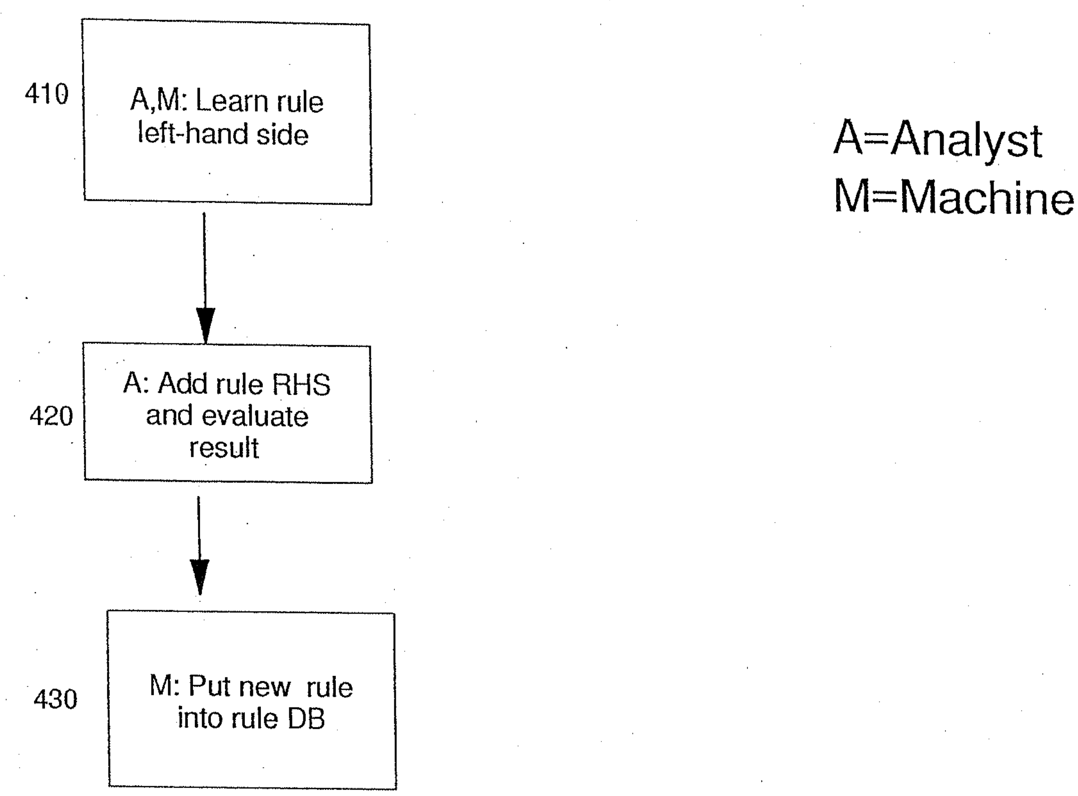

410
A,M: Learn rule left-hand side
420
A: Add rule RHS and evaluate result
430
M: Put new rule into rule DB
A=Analyst
M=Machine

Fig. 3

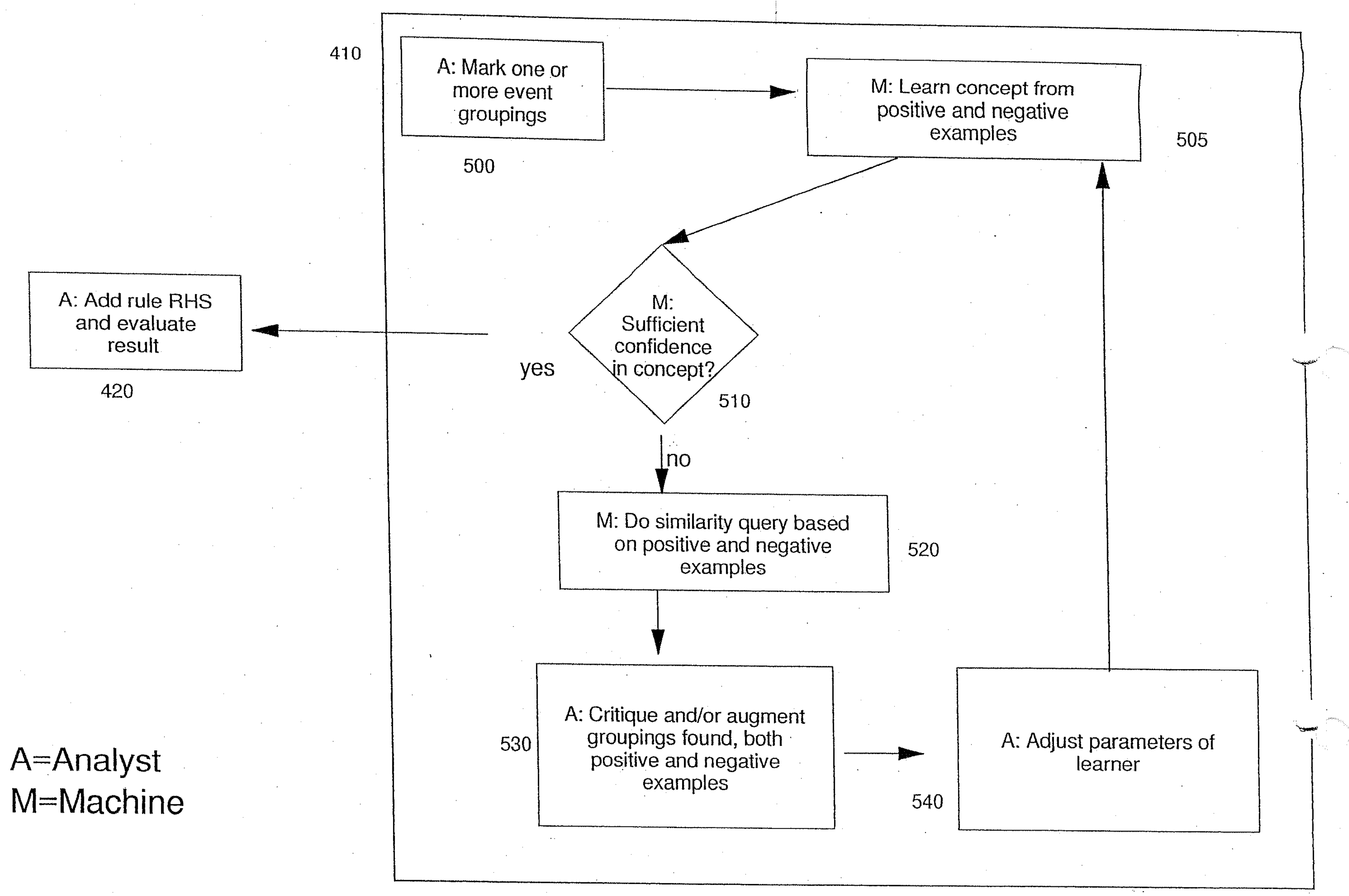

410
A: Mark one or more event groupings
500
M: Learn concept from positive and negative examples
505
M: Sufficient confidence in concept?
510
yes
A: Add rule RHS and evaluate result
420
no
M: Do similarity query based on positive and negative examples
520
A: Critique and/or augment groupings found, both positive and negative examples
530
A: Adjust parameters of learner
540
A=Analyst
M=Machine

Fig. 4

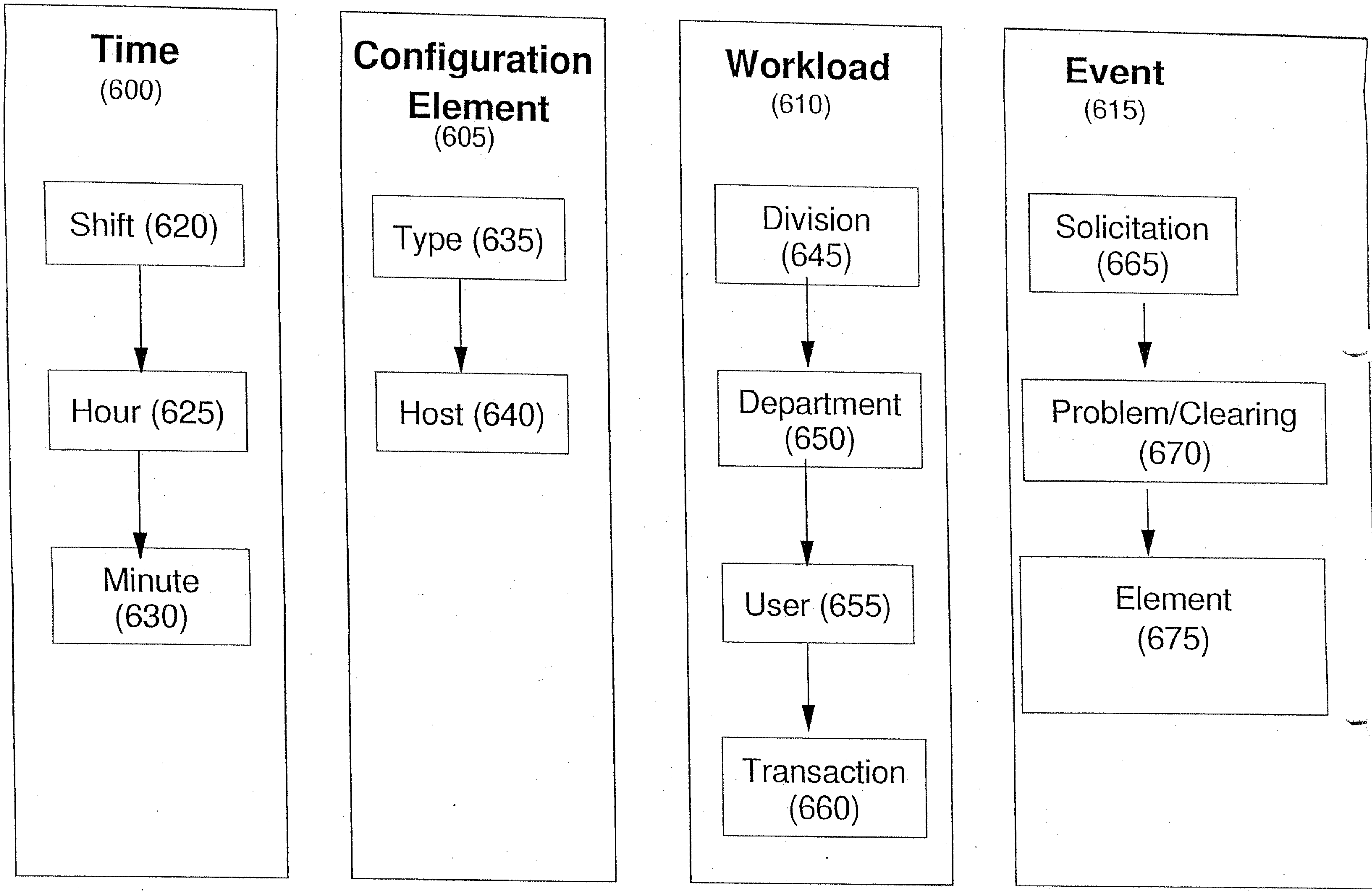
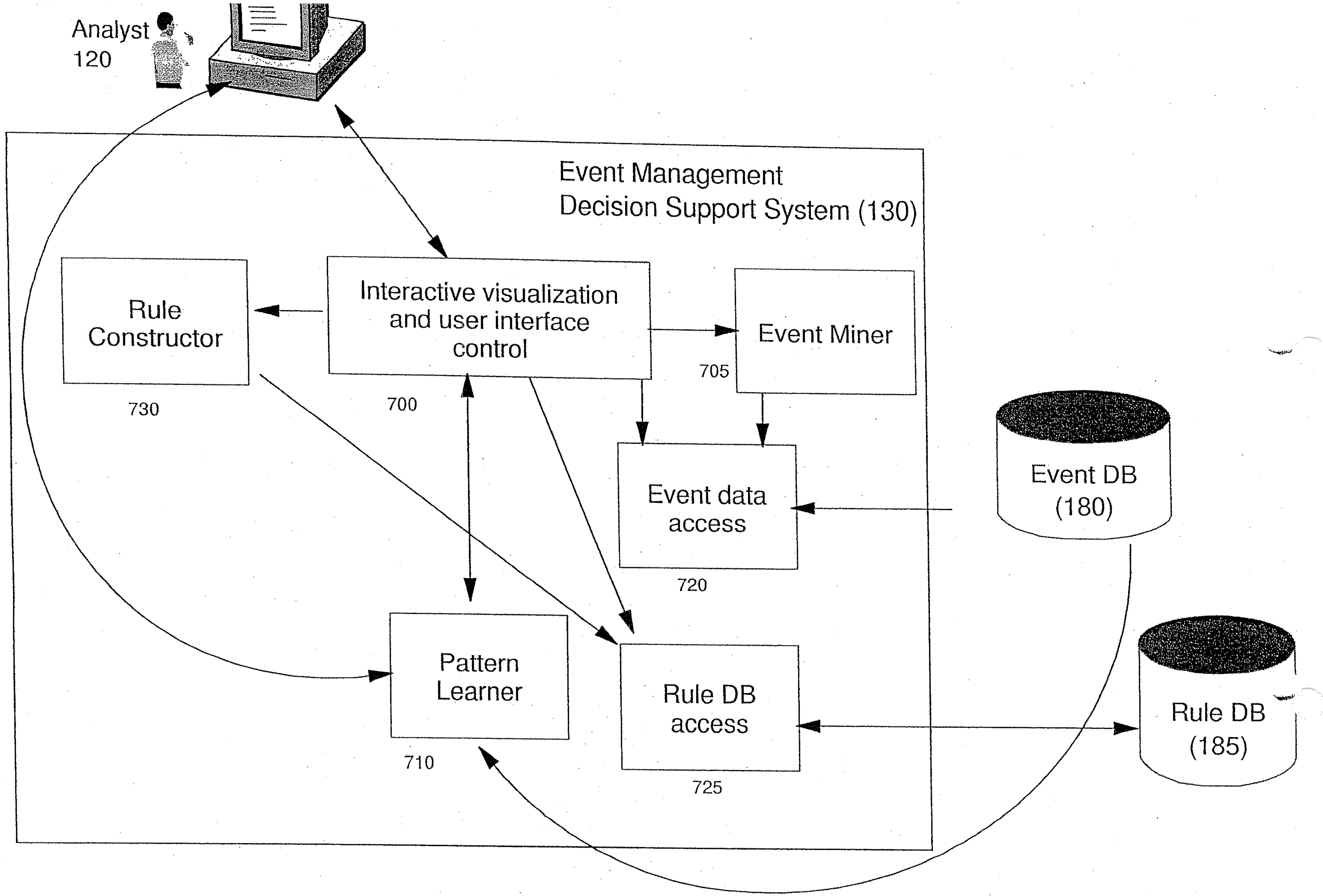

Time
(600)
Shift (620)
Hour (625)
Minute
(630)
Configuration
Element
(605)
Type (635)
Host (640)
Workload
(610)
Division
(645)
Department
(650)
User (655)
Transaction
(660)
Event
(615)
Solicitation
(665)
Problem/Clearing
(670)
Element
(675)

Fig. 5



Analyst
120
Event Management
Decision Support System (130)
Rule
Constructor
730
Interactive visualization
and user interface
control
700
Event Miner
705
Event data
access
720
Pattern
Learner
710
Rule DB
access
725
Event DB
(180)
Rule DB
(185)

Fig. 6

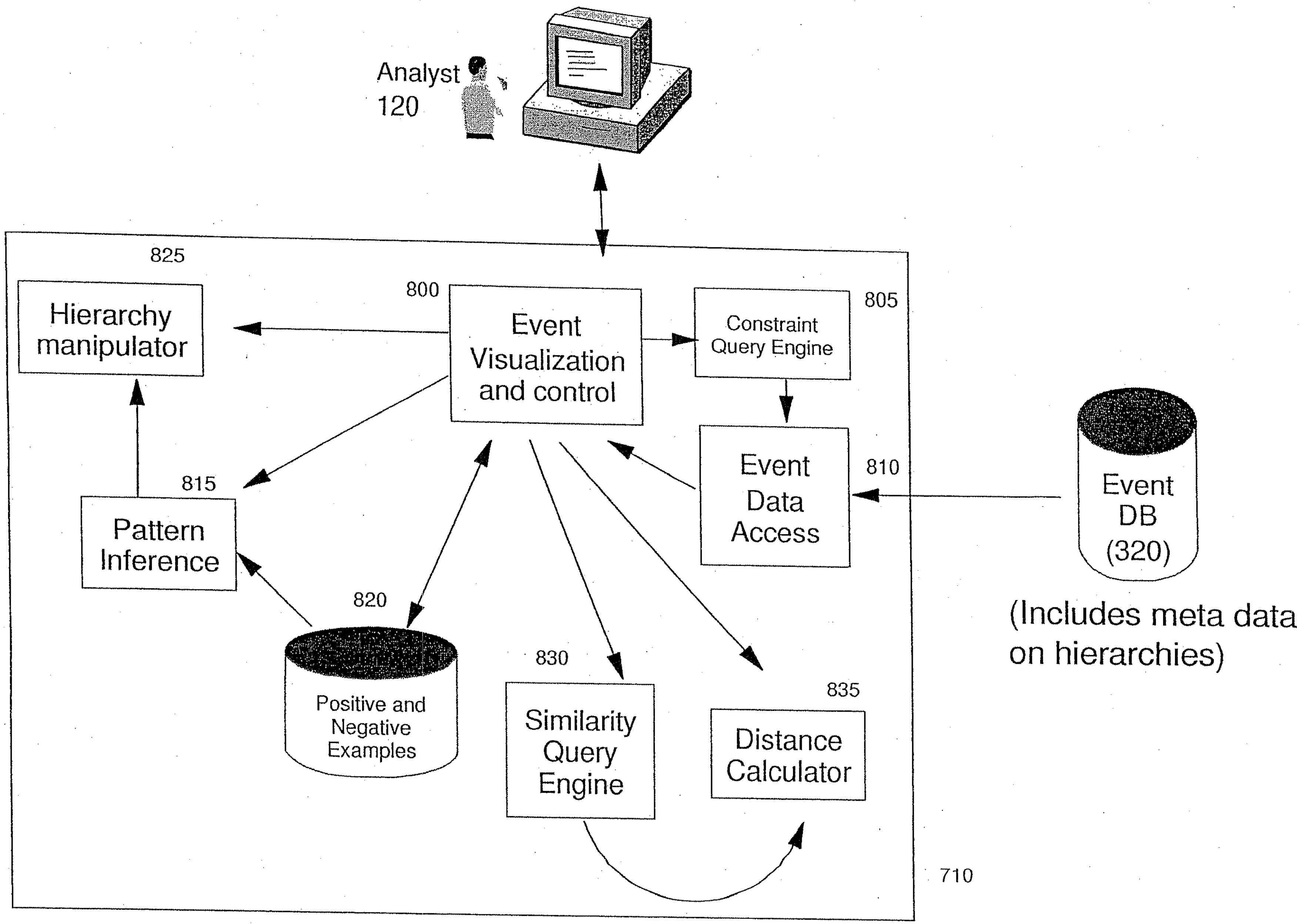

Analyst
120
825
Hierarchy manipulator
800
Event Visualization and control
805
Constraint Query Engine
810
Event Data Access
Event DB (320)
(Includes meta data on hierarchies)
815
Pattern Inference
820
Positive and Negative Examples
830
Similarity Query Engine
835
Distance Calculator
710

Fig. 7

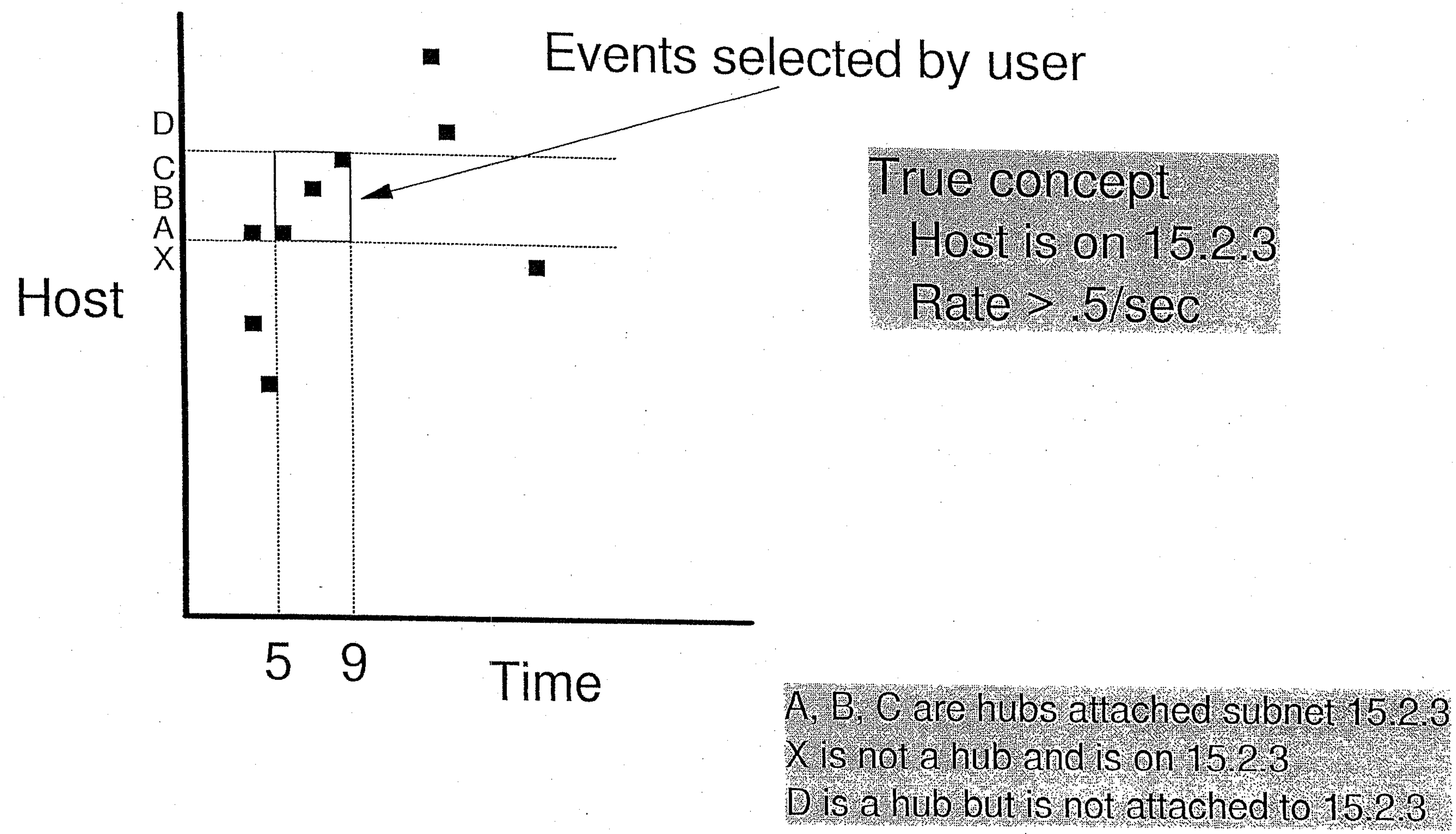
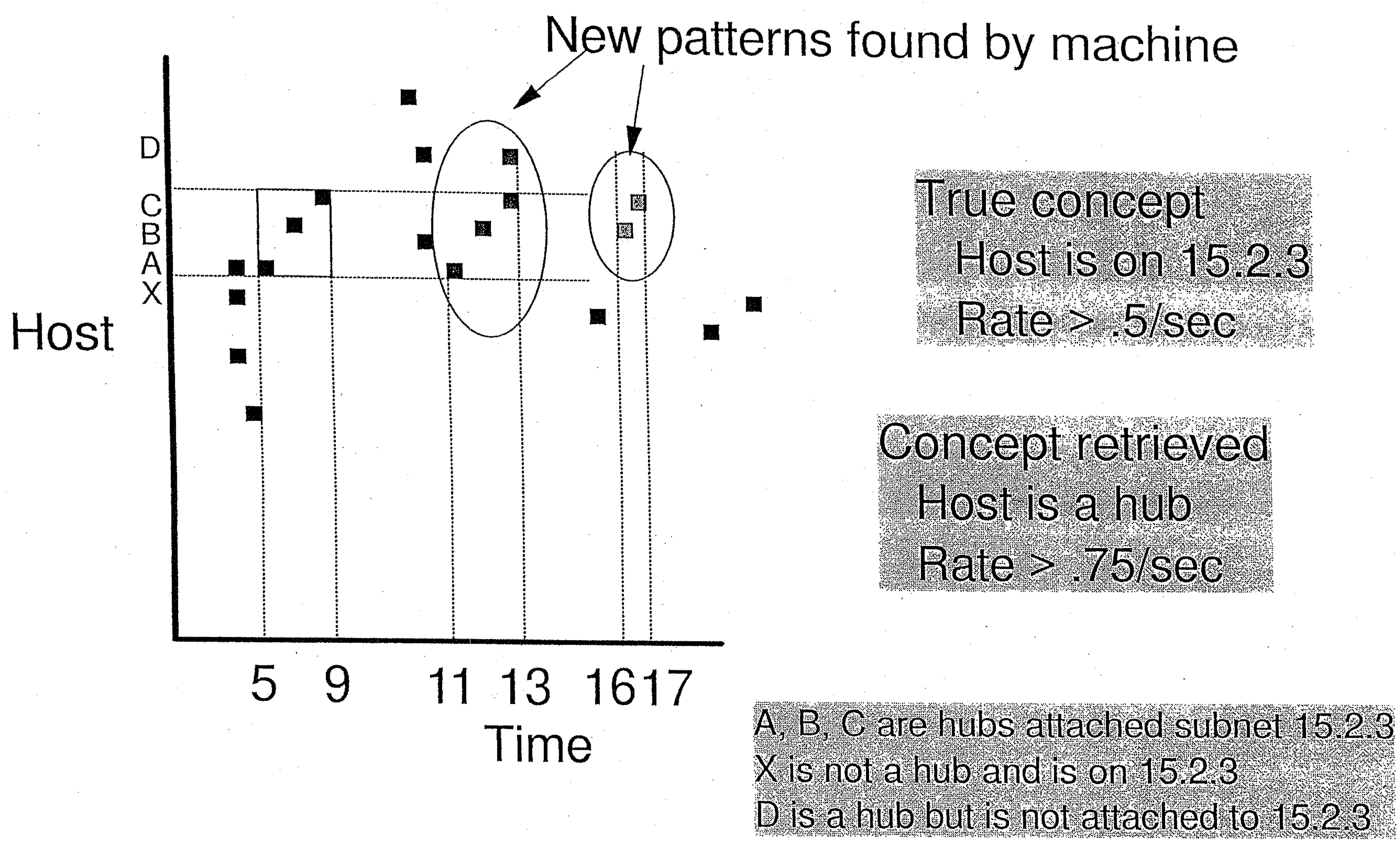

Events selected by user
True concept
Host is on 15.2.3
Rate > .5/sec
Host
D
C
B
A
X
5
9
Time
A, B, C are hubs attached subnet 15.2.3
X is not a hub and is on 15.2.3
D is a hub but is not attached to 15.2.3

Fig. 8



New patterns found by machine
Host
D
C
B
A
X
5
9
11
13
16
17
Time
True concept
Host is on 15.2.3
Rate > .5/sec
Concept retrieved
Host is a hub
Rate > .75/sec
A, B, C are hubs attached subnet 15.2.3
X is not a hub and is on 15.2.3
D is a hub but is not attached to 15.2.3

Fig. 9

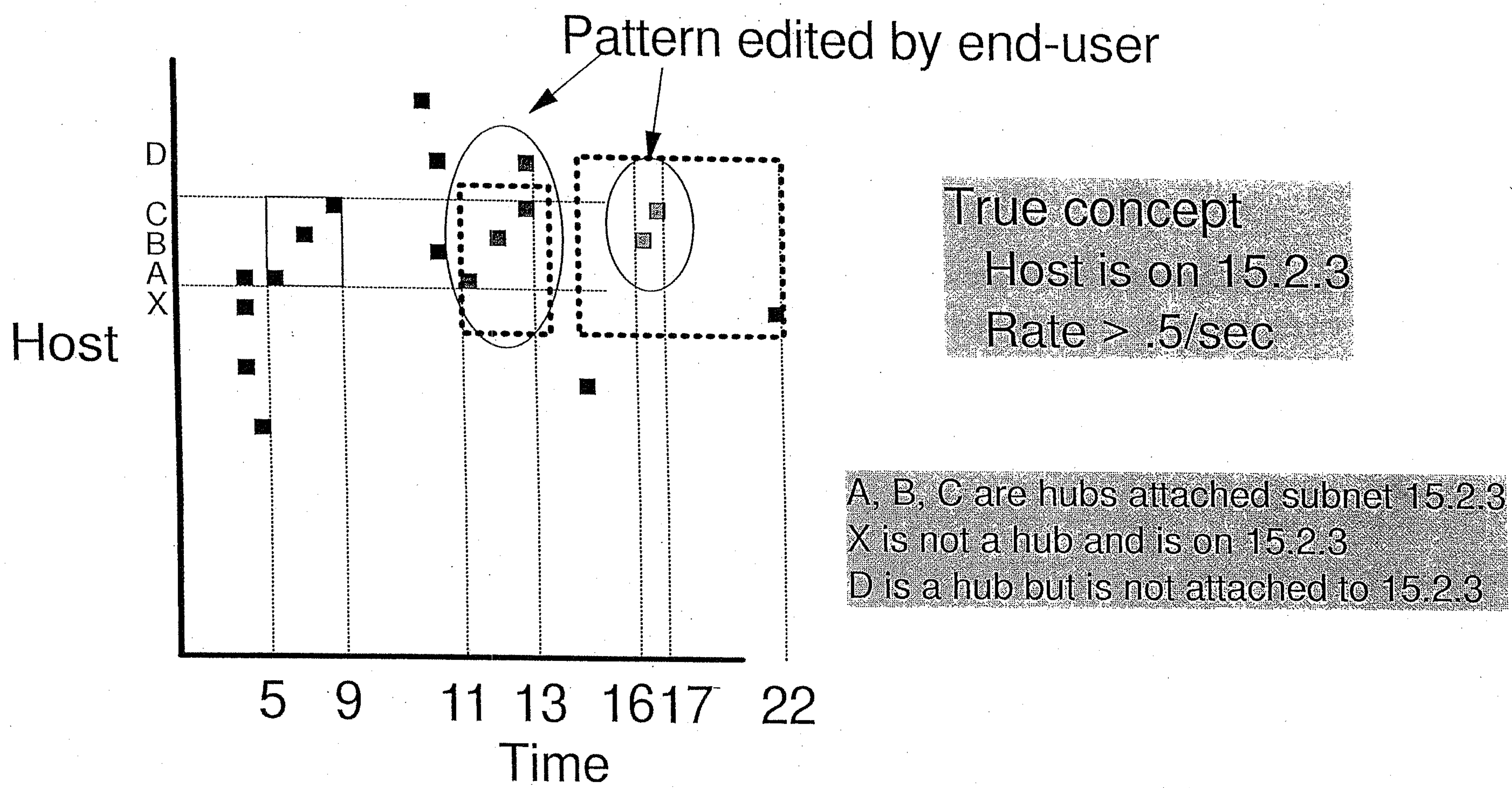

Pattern edited by end-user
Host
D
C
B
A
X
5
9
11
13
16
17
22
Time
True concept
Host is on 15.2.3
Rate > .5/sec
A, B, C are hubs attached subnet 15.2.3
X is not a hub and is on 15.2.3
D is a hub but is not attached to 15.2.3

Fig. 10

EXHIBIT 2

RYAN, MASON & LEWIS, LLP
ATTORNEYS AT LAW
90 FOREST AVENUE
LOCUST VALLEY, NEW YORK 11560
Telephone: (516) 759-2946
Facsimile: (516) 759-9512
Email: wel@rml-law.com

DATE: November 27, 2000	**FILE:** YOR920000581US1

Facsimile Message From: **WILLIAM E. LEWIS**

Please deliver the following pages to:

NAME: Joseph L. Hellerstein
OF: IBM Corporation
FAX NUMBER: (914) 784-6040
NUMBER OF PAGES INCLUDING THIS COVER PAGE: 33
COMMENTS/INSTRUCTIONS:

Dear Joe:

Please find attached a draft of the patent application relating to your invention. I kindly ask that you review the attached draft, and have your co-inventors do the same, and provide me with your comments at your earliest convenience. Please confirm receipt. Thank you for your assistance.

Best regards,

Bill Lewis

Bill Lewis

If you do not receive all of the pages, please call us back as soon as possible at (516) 759-2946.

EXHIBIT 3

RYAN, MASON & LEWIS, LLP
ATTORNEYS AT LAW
90 FOREST AVENUE
LOCUST VALLEY, NEW YORK 11560
Telephone: (516) 759-2946
Facsimile: (516) 759-9512
Email: wel@rml-law.com

DATE: December 4, 2000	**FILE:** YOR920000581US1

Facsimile Message From: **WILLIAM E. LEWIS**

Please deliver the following pages to:

NAME: Joseph L. Hellerstein

OF: IBM Corporation

FAX NUMBER: (914) 784-6040

NUMBER OF PAGES INCLUDING THIS COVER PAGE: 36

COMMENTS/INSTRUCTIONS:

Dear Joe:

Please find attached a revised draft of the patent application relating to your invention. The revised draft incorporates your recent comments. I kindly ask that you review the attached revised draft in its entirety, and have your co-inventors do the same, and provide me with your comments as soon as possible. As you know, we plan to file this application on or before **Friday, December 8, 2000**. Please confirm receipt. Thank you again for your assistance.

Best regards,

Bill

Bill Lewis

If you do not receive all of the pages, please call us back as soon as possible at (516) 759-2946.

PTO/SB/96 (08-00)
Approved for use through 10/31/2002. OMB 0651-0031
U.S.Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

STATEMENT UNDER 37 CFR 3.73(b)

Applicant/Patent Owner: J.L. Hellerstein et al.

Application No./Patent No.: 09/731,937 Filed/Issue Date: December 7, 2000

Entitled: Method and System for Machine-Aided Rule Construction for Event Management

International Business Machines Corporation, a corporation,
(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that it is:

1. ☑ the assignee of the entire right, title, and interest; or

2. ☐ an assignee of less than the entire right, title and interest.
The extent (by, percentage) of its ownership interest is _______%

in the patent application/patent identified above by virtue of either:

A. ☑ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel 011683, Frame 0337, or for which a copy thereof is attached.

OR

B. ☐ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as shown below:

1. From:______________ To:______________
The document was recorded in the United States Patent and Trademark Office at Reel __________, Frame__________, or for which a copy thereof is attached.

2. From:______________ To:______________
The document was recorded in the United States Patent and Trademark Office at Reel __________, Frame__________, or for which a copy thereof is attached.

3. From:______________ To:______________
The document was recorded in the United States Patent and Trademark Office at Reel __________, Frame__________, or for which a copy thereof is attached.

Additional documents in the chain of title are listed on a supplemental sheet.

☒ Copies of assignments or other documents in the chain of title are attached.
[NOTE: A separate copy (*i.e.*, the original assignment document or a true copy of the original document) must be submitted to Assignment Division in accordance with 37 CFR Part 3, if the assignment is to be recorded in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

9-20-05
Date

Lynne D. Anderson
Typed or printed name

[signature]
Signature

Program Manager, Patent Office Liaison
Title

LETTER OF AUTHORITY

I, Gerald Rosenthal, Vice President, Intellectual Property and Licensing of International Business Machines Corporation (IBM), a New York corporation, do hereby delegate the authority to approve and execute documents on behalf of IBM relating to proceedings in the Patent, Trademark Registration or Copyright Offices servicing any country or region of the world, or to related appeal proceedings, including, but not limited to: petitions; powers of Attorney; authorizations; verification; nominations of representatives; declarations; documents relating to maintenance and defense of the resulting industrial property rights; assignments of rights to apply for and acquire patents and trademark registrations; and evidence of such assignments; requests for the registration of patents as available for licensing; reports of inventions and petitions for waiver of patent rights to any department or agency of the United States Government; and, assignments, licenses and other instruments confirmatory of Government rights in patents and inventions, to Lynne D. Anderson, Program Manager, Patent Office Liaison.

Date: April 6, 2004

Gerald Rosenthal
Vice President, Intellectual Property & Licensing
International Business Machines Corporation

EXHIBIT C

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s):	J.L. Hellerstein et al.
Case:	YOR920000581US1
Serial No.:	09/731,937
Filing Date:	December 7, 2000
Group:	2144
Examiner:	Thanh T. Nguyen

Title: Method and System for Machine-Aided Rule Construction for Event Management

---

ATTORNEY AFFIDAVIT

I, the undersigned, hereby declare and state as follows:

1. I prepared and filed the above-referenced U.S. patent application.

2. On or about August 28, 2000, I received an internal IBM invention disclosure document entitled "Method and System for Machine-Aided Rule Construction for Event Management," with instructions to prepare and file a U.S. patent application based on the above-mentioned document. A copy of the accompanying letter is attached hereto as Exhibit 1.

3. Between August 28, 2000 and November 27, 2000, I had a reasonable backlog of cases which were taken up in chronological order and carried out expeditiously before working reasonably hard on and completing a draft patent application for the above-referenced U.S. patent application.

4. On or about November 27, 2000, I sent a draft patent application to inventor Joseph Hellerstein for review by the inventors. The fax transmission cover sheet is attached hereto as Exhibit 2.

5. On or about December 4, 2000, I sent a revised draft patent application to inventor Joseph Hellerstein after receiving comments from the inventors and revising the draft patent application. The fax transmission cover sheet is attached hereto as Exhibit 3.

6. The invention was constructively reduced to practice by filing the above-referenced patent application on December 7, 2000, after inventors reviewed and approved the revised draft patent application.

6. All statements made herein of my own knowledge are true, and all statements made on information and belief are believed to be true.

7. I understand that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001, and may jeopardize the validity of the application or any patent issuing thereon.

Date: 9/19/05

William E. Lewis
William E. Lewis
Attorney for Applicant(s)
Reg. No. 39,274

EXHIBIT 1

IBM

August 24, 2000

Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

William E. Lewis, Esq.
Ryan & Mason, LLP
90 Forest Avenue
Locust Valley, New York 11560

RECEIVED
AUG 2 8 2000
BY: J.E.R.

Subject: Preparation of Patent Application: YOR920000581US1
Yorktown Disclosure Number: YOR8-2000-0309

RECEIVED
WITH THANKS
RYAN & MASON, L.L.P
J.E.R.

Title: " METHOD AND SYSTEM FOR MACHINE-AIDED RULE CONSTRUCTION FOR EVENT MANAGEMENT"

-162

Inventor:	Joseph L. Hellerstein	(914) 784-7506	2S-D42	Hawthorne
	Sheng Ma	(914) 784-7837	H2-K22	Hawthorne
	David A. Rabenhorst	(914) 784-7638	4S-B34	Hawthorne

Dear Bill:

Further to our telephone conversation of last week, enclosed are materials relative to the preparation and prosecution of the subject patent application including an original disclosure, an embodiment, a diskette with a soft copy of the embodiment text in Lotus Word Pro format and figures in Lotus Freelance format and a paper to be published in December. The patent application is to be filed in the USPTO **on or before November 24, 2000**, due to an impending bar. *Please add a Beauregard claim if applicable*. The work is to be done in accordance with the IBM Outside Counsel Instructions.

The formal papers are to be prepared and filed by your office, listing the names of the Yorktown attorneys on the Declaration and Power of Attorney as follows:

POWER OF ATTORNEY: As a named inventor I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith as follows:

Manny W. Schecter (Reg. 31,722), Lauren C. Bruzzone (Reg. 35,082), Christopher A. Hughes (Reg. 26,914), Edward A. Pennington (Reg. 32,588), John E. Hoel (Reg. 26,279), Joseph C. Redmond, Jr. (Reg. 18,753), Richard M. Ludwin (Reg. 33,010), Marc A. Ehrlich (Reg. 39,366), Douglas W. Cameron (Reg. 31,596), Stephen C. Kaufman (Reg 29,551), Marian Underweiser (Reg. 46,134), Wayne L. Ellenbogen (Reg. 43,602), Robert M. Trepp (Reg 25,933), Louis P. Herzberg (Reg. 41,500), Louis J. Percello (Reg. 33,206), Paul J. Otterstedt (Reg. 37,411), Daniel P. Morris (Reg. 32,053), and David M. Shofi (Reg. 39, 835).

Send correspondence to: outside counsel attorney
Direct Telephone Calls to: outside counsel attorney

When filling out the Assignment, please include a notarization sheet.

After the formal papers have been prepared by your office, please send them directly to the inventors for signature with a request that they sign and have the papers

notarized. Please instruct the inventors to return the executed formal papers directly to your office.

Please be advised that an additional step in our procedure is required when filing all original IBM Yorktown Patent Applications in the USPTO. The additional step is that a "Taiwan Oath & Assignment" form must be completed. The form must have all the required information completely filled in and must be signed and dated by all named inventors in the subject patent application. The second page of the two page form has a section entitled, "Note 1", which contains an alphabetized code reference depicting what information must be entered into each corresponding letter field (example: (a), (b), etc.). Please use the diskette which was sent to you for the soft copy of the "Taiwan Oath & Assignment" form. When the form is complete, and all inventor(s) signatures and dates have been obtained, please forward the original Taiwan Oath & Assignment back to my office.

An Associate Power of Attorney form should be prepared and forwarded to this office for signature.

Before filing, please send me, by facsimile or otherwise, a copy of the claims if not the application as a whole, for my review. Furthermore, please send me a copy of the application (hard-copy and soft-copy (Word Pro 97 preferable)) and the formal papers, as filed, by express mail concurrently with your filing of the application. IBM will handle any US Maintenance fee payments internally. (Please note that the fee addressee should be our office in Yorktown Heights.)

Please conduct the work directly with the inventor listed above, including visiting as the situation warrants. The contact inventor is Joseph Hellerstein.

Please inform this office on all points involving scope of coverage and finances. Please have your illustrator prepare formal drawings for the application. In the event you file with informal drawings, please provide us with formal drawings within three months of the filing date. If you have any questions contact me at the telephone number listed below.

Best regards,

David M. Shofi
(914) 945-3247

/mm
Enclosures
cc: L. Bruzzone
B. Rasa
J. Hellerstein
S. Ma
D. Rabenhorst

EXHIBIT 2

RYAN, MASON & LEWIS, LLP
ATTORNEYS AT LAW
90 FOREST AVENUE
LOCUST VALLEY, NEW YORK 11560
Telephone: (516) 759-2946
Facsimile: (516) 759-9512
Email: wel@rml-law.com

DATE: November 27, 2000	**FILE:** YOR920000581US1

Facsimile Message From: **WILLIAM E. LEWIS**

Please deliver the following pages to:

NAME:	Joseph L. Hellerstein
OF:	IBM Corporation
FAX NUMBER:	(914) 784-6040
NUMBER OF PAGES INCLUDING THIS COVER PAGE:	33
COMMENTS/INSTRUCTIONS:	

Dear Joe:

Please find attached a draft of the patent application relating to your invention. I kindly ask that you review the attached draft, and have your co-inventors do the same, and provide me with your comments at your earliest convenience. Please confirm receipt. Thank you for your assistance.

Best regards,

Bill Lewis

Bill Lewis

If you do not receive all of the pages, please call us back as soon as possible at (516) 759-2946.

EXHIBIT 3

RYAN, MASON & LEWIS, LLP
ATTORNEYS AT LAW
90 FOREST AVENUE
LOCUST VALLEY, NEW YORK 11560
Telephone: (516) 759-2946
Facsimile: (516) 759-9512
Email: wel@rml-law.com

DATE: December 4, 2000	**FILE:** YOR920000581US1

Facsimile Message From: **WILLIAM E. LEWIS**

Please deliver the following pages to:

NAME:	Joseph L. Hellerstein
OF:	IBM Corporation
FAX NUMBER:	(914) 784-6040
NUMBER OF PAGES INCLUDING THIS COVER PAGE:	36

COMMENTS/INSTRUCTIONS:

Dear Joe:

Please find attached a revised draft of the patent application relating to your invention. The revised draft incorporates your recent comments. I kindly ask that you review the attached revised draft in its entirety, and have your co-inventors do the same, and provide me with your comments as soon as possible. As you know, we plan to file this application on or before **Friday, December 8, 2000**. Please confirm receipt. Thank you again for your assistance.

Best regards,

Bill

Bill Lewis

If you do not receive all of the pages, please call us back as soon as possible at (516) 759-2946.

RELATED PROCEEDINGS APPENDIX

None